ON IMPROVED LEAST SQUARES REGRESSION & ARTIFICIAL NEURAL NETWORK META-MODELS FOR SIMULATION VIA CONTROL VARIATES

DISSERTATION

Michael P. Gibb, Captain, USAF

AFIT–ENS–DS–16–S–030

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT–ENS–DS–16–S–030

ON IMPROVED LEAST SQUARES REGRESSION & ARTIFICIAL NEURAL NETWORK METAMODELS FOR SIMULATION VIA CONTROL VARIATES

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy in Operations Research

Michael P. Gibb, B.S., M.S.

Captain, USAF

September 2016

AFIT–ENS–DS–16–S–030

ON IMPROVED LEAST SQUARES REGRESSION & ARTIFICIAL NEURAL NETWORK METAMODELS FOR SIMULATION VIA CONTROL VARIATES

Michael P. Gibb, B.S., M.S.
Captain, USAF

Committee Membership:

Kenneth W. Bauer, Jr., Ph.D.
Chair

John O. Miller, Ph.D.
Member

Brian J. Lunday, Ph.D.
Member

Edward D. White, Ph.D.
Member

ADEDEJI B. BADIRU, Ph.D.
Dean, Graduate School of Engineering
and Management

Abstract

The analysis of large-scale simulations can pose a large computational burden, often necessitating the use of high performance computers. Moreover, these models can be analytically intractable because of complex, internal logical relationships. To begin to overcome these types of obstacles, a method known as *meta-modeling* has been developed to construct mathematical functions that act as analytic surrogates to large scale simulations. This research examines the introduction of second-order interactions for two types of asymptotically-standardized linear control variates to least squares regression and radial basis neural network meta-models for a queuing simulation. Extensions are made to the statistical framework for variance reduction of direct estimation of single response, single population simulations and the framework for meta-models of single response, multiple population simulations. As a result, the new extensions are shown to significantly outperform existing frameworks and also provide the means to interpret and better understand the system dynamics that manifest when a system exhibits an exceptional amount of variance.

Ἰησοῦς Χριστός

Behold the Lamb of God, which taketh away the sin of the world.

– John 1:29 –

To my wife

Though I am old with wandering

Through hollow lands and hilly lands,

I will find out where she has gone

And kiss her lips and take her hands;

And walk among long dappled grass,

And pluck till time and times are done

The silver apples of the moon,

The golden apples of the sun.

– W.B. Yeats –

To my children

On the day that you were born the angels got together

And decided to create a dream come true

So they sprinkled moon dust in your hair

Of gold and starlight in your eyes of blue

– The Carpenters –

Acknowledgements

Foremost, I give all glory to God for his timely provisions and for equipping me with the necessary reasoning and determination to complete this dissertation. *AMDG*

I thank the U.S. Air Force for sponsoring this opportunity to an aspiring former Staff Sergeant who as a child completed third grade twice. I would like to express my utmost respect and appreciation to my research advisor, Dr. Kenneth W. Bauer, Jr.. His astute guidance helped me overcome many moments of crisis. I will fondly remember hearing frequently that, "I live and die by the computer". I also acknowledge the high standards and constructive feedback from the esteemed members of my research committee: Dr. John O. Miller, Dr. Brian J. Lunday, and Dr. Edward D. White. I also give special thanks to my friend and colleague, Maj. Joe Bellucci, Ph.D.

I extend a heartfelt tribute to my entire family, acutely so to my father, for whom I'm exceptionally proud of his heritage – a testament of his hard labor; my mother, who auspiciously dedicated much of her attention to my education when I was a child; and my sister, for sharing her enthusiasm and counsel in my life.

A profound recognition is due to my marvelous, resilient children who provided a tremendous amount of love, laughter, and respite from my scholarly diversions. I pray this endeavor will serve as an encouragement for their future academic pursuits.

Finally, my efforts alone would have remained fruitless, if not for the invaluable contributions from my darling, steadfast wife. The significance of my prolonged residence in the ivory towers of academia vanishes beyond her enchanting presence.

Michael P. Gibb

Table of Contents

Page

Abstract iv

Dedication v
Acknowledgements vi

List of Figures xi

List of Tables xiv

I. Introduction & Literature Review 1

1.1 Background 1
1.2 Statistical Framework 4
1.2.1 Control Variates (CV) 4
1.2.1.1 Single Control Variate (CV) for Single Population, Single Response Simulation 8
1.2.1.2 Multiple Control Variates for Single Population, Single Response Simulation 11
1.2.1.3 Multiple Control Variates for Single Population, Multiple Response Simulation 13
1.2.1.4 Control Variate Selection 16
1.2.1.5 Internal Control Variates 17
1.2.1.6 External Control Variates 18
1.2.1.7 Combination of Control Variates 18
1.2.1.8 Standardized "Work" Control Variates (CVs) 19
1.2.1.9 Standardized "Routing" Control Variates (CVs) 20
1.2.1.10 Method of Replication with Control Variates (CVs) 21
1.2.1.11 Method of Direct Simulation 22
1.2.2 Neural Networks 23
1.2.2.1 Fundamentals of Radial Basis Neural Network (RBNN) 28
1.2.2.2 Design of Experiments with Neural Nets 31
1.2.2.3 Factorial Design 32
1.2.2.4 Latin Hypercube Design 33
1.2.2.5 Central Composite Design 34
1.2.2.6 Other designs 35
1.2.3 Simulation Meta-models 35
1.2.3.1 Meta-model Fundamentals 37

Page

1.2.3.2 General Linear Regression Meta-model . . . 39
1.2.3.3 Multiple Linear Regression Meta-model . . . 40
1.2.3.4 Multivariate Linear Regression Model . . . 41
1.2.3.5 Least Squares Optimality . . . 44
1.2.3.6 Regression Meta-model with Control Variates . . . 45
1.2.3.7 Multiple Control Variates for Multiple Population, Single Response Simulation . . . 46
1.2.3.8 Multiple Control Variates for Multiple Population, Multiple Response Simulation . . . 47
1.2.3.9 Neural Network Meta-model . . . 49
1.2.4 Jackson Network Analysis for Open Networks with M/M/s/∞ queues . . . 52
1.2.5 Interior Point method for Non-linear Convex Optimization . . . 54
1.2.6 TV Inspection and Adjustment Stations on a Production Line Simulation for Research Objectives . . . 58
1.3 Research Overview . . . 59

II. Unveiling System Dynamics for Single Population Simulations Using Linear Control Variates with Interactions . . . 60

2.1 Introduction . . . 60
2.2 Control Variate Model with Interactions . . . 62
2.3 Case Study . . . 64
2.3.1 TV Inspection and Adjustment Production System . . . 64
2.3.2 Part I: Variance Reduction with Non-terminating Simulation . . . 65
2.3.3 Part II: Unveiling System Dynamics with Non-terminating Simulation . . . 75
2.3.3.1 Experiment 1: Replicated Linearly Regressed Coefficients . . . 76
2.3.3.2 Experiment 2: Linearly Regressed, Statistically Significant Coefficients w/ Max Replications . . . 79
2.4 Conclusions . . . 83

III. Improved Multi-Population Least Squares Regression Metamodels for Simulations Using Linear Control Variates with Interactions . . . 84

Page

3.1 Introduction . . . 84
3.2 Second Order Interaction Metamodel with Multiple Control Variates for a Multiple Population, Single Response Simulation . . . 85
3.3 Case Study . . . 88
3.3.1 Simulation Analysis . . . 88
3.3.2 Non-linear Transformation of Simulation Responses . . . 94
3.3.3 Meta-Experimentation for Least Squares Regression Meta-models . . . 98
3.4 Results . . . 105
3.4.1 Second order Least Squares Regression Meta-model with CV Interaction Performance . . . 105
3.4.2 System Dynamics . . . 111
3.5 Conclusions . . . 115

IV. Improved Neural Network Metamodels for Simulations Using Linear Control Variates . . . 117

4.1 Introduction . . . 117
4.2 Extended Radial Basis Neural Network with CVs . . . 117
4.3 Case Study . . . 119
4.3.1 Simulation Analysis . . . 121
4.3.2 Approximate Optimal Radial Basis Neural Network Spread & Neuron Architecture Heuristic . . . 124
4.3.3 Meta-Experimentation for Radial Basis Neural Network Meta-models . . . 131
4.4 Results . . . 136
4.5 Conclusion . . . 142

V. Conclusion . . . 143

5.1 Summary . . . 143
5.2 Original Contributions . . . 143
5.2.1 Increased Variance Reduction for Direct Estimation of Single Response, Single Population Simulations . . . 143
5.2.2 Unveiling System Dynamics for a Single Response, Single Population Simulations . . . 144
5.2.3 Improved Least Squares Regression Meta-model for Single Response, Multiple Population Simulations . . . 144

Page

5.2.4 System Dynamics for Single Response, Multiple Population Simulations . . . 145
5.2.5 Improved Radial Basis Neural Network Meta-model for Single Response, Multiple Population Simulations . . . 146
5.3 Future Research . . . 146
5.4 Final Remarks . . . 147

A. Additional Tables and Figures . . . 148

A.1 Information for 73 Simulation Configurations . . . 148
A.2 Additional Table for Chapter I: Single Population w/ 2nd Order CVs . . . 152
A.3 Additional Figures: Simulation / Jackson Formula Projections . . . 154
A.4 Estimating Coefficients for $2LSR_{CV}$ meta-model . . . 155
A.5 Chapter III: Least Squares Regression Meta-model Figures . . . 156
A.6 Chapter IV: RBNN Meta-model Figures . . . 160

B. Extended Literature Review . . . 164

B.1 Single Control Variate (CV) for Single Population, Single Response Simulation . . . 164
B.2 Artificial Feed-forward Neural Networks . . . 170
B.3 Least Squares Optimality . . . 178
B.4 Unbiased Quadratic Control Variates . . . 178
Bibliography . . . 188

List of Figures

Figure		Page
1	Sketch of biological neuron. [50]	24
2	McCulloch illustrated network of nodes. [76]	24
3	Linear Threshold Function (Left) and McCulloch & Pitts model (Right) [76] [50]	25
4	Types of activation functions. [50]	27
5	Taxonomy of neural net architectures. [50]	27
6	Example Radial Basis Neural Network (RBNN) *excluding* Standardized "Work" and "Routing" Control Variates	29
7	Example 2^3 factorial design [84]	33
8	Example Latin Hypercube design with 10 points [80]	34
9	Example Central Composite Design with center and axial points [79]	35
10	Simulation meta-model in a Decision Support System [71]	37
11	Two Dimensional Convexity Example	55
12	Two Dimensional Non-Convexity Example	55
13	Diagram of TV Inspection & Adjusting System [16]	58
14	Absolute vale of estimating coefficient for 100-1,000 replications	77
14	Simulation / Jackson Formula Projections for All 73 Design Points	93
15	Non-linear transformations of Y^{coded} and U_A^{coded}	96
16	Non-linear transformations of interaction terms between Y^{coded} and U_A^{coded}	97
17	Surface Plots of Four Nonlinear Cost Function Transformations	98

Figure Page

18 Abstract Flowchart for Multiple Population Regression Meta-experiment 104

19 Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time 107

20 Design: FF - Full Factorial, Response: CF_2 - Cost Function 2 108

21 Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time 109

22 Design: FF - Full Factorial, Response: CF_2 - Cost Function 2 109

23 Design: FF - Full Factorial 110

24 Example Extended Radial Basis Neural Network with CVs (RB_{CV}) 119

25 Abstract Flowchart for Approximate Optimal RB_CV Spread & Neuron Architecture Heuristic 125

26 Results for RB_{CV} Assumptions Experiment, Design: FF - Full Factorial, Response: CF_2 - second Cost Function 128

27 Abstract Flowchart for Radial Basis Neural Network Meta-experiment 136

28 Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time 138

29 Design: FF - Full Factorial, Response: CF_2 - Cost Function 2 139

30 Design: FF - Full Factorial 141

31 Simulation / Jackson Formula Projections for Five Designed Experiment 154

32 Design: CC_f - Faced Central Composite 156

33 Design: CC_c - Circumscribed Central Composite 157

Figure Page

34 Design: BB - Box-Behnken 158

35 Design: LH - Latin Hypercube 159

36 Design: CC_f - Faced Central Composite 160

37 Design: CC_c - Circumscribed Central Composite 161

38 Design: BB - Box-Behnken 162

39 Design: LH - Latin Hypercube 163

40 Single perceptron with bias 171

41 Multilayer perceptron artificial neural network with bias 172

List of Tables

Table		Page
1	Taxonomy of Control Variate Publications	7
2	Average TV sojourn time in system	66
3	Subsets of CVs ($X \equiv$ term included)	67
4	Realized coverage [%]	69
5	Average variance reduction relative to confidence interval w/o CVs [%]	71
6	Average variance reduction relative to standard error of the confidence interval w/o CVs [%]	73
7	Estimating coefficients for WR models using 1,000 replications	80
8	Estimating coefficients for WRi models using 1,000 replications	81
9	Range of Simulation Input Settings and Output Ranges	89
10	Description of Simulation Control Variates	90
11	Description of Designed Experiments	94
12	Description of K fold method	101
13	Covariance of the FF, Full Factorial Designed Input Variates, X, and the Standardized "Work" and "Routing" Control Variates, W and R	112
14	Regression Coefficients from All Meta-models for FF & CF_2, where $p > 0$	114
15	Approximate Optimal Radial Basis Neural Network including CVs (RB_{CV}) Architecture	130
16	Approximate Optimal Radial Basis Neural Network excluding CVs (RB_{noCV}) Architecture	131

Table Page

17 Designed Settings (1-20) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Long Run Mean Responses 148

18 Designed Settings (21-40) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses 149

19 Designed Settings (41-60) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses 150

20 Designed Settings (61-73) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses 151

21 Average variance reduction relative to standard error of the confidence interval w/o CVs [%] 152

22 Average variance reduction relative to standard error of the confidence interval w/o CVs [%] 153

ON IMPROVED LEAST SQUARES REGRESSION & ARTIFICIAL NEURAL NETWORK METAMODELS FOR SIMULATION VIA CONTROL VARIATES

I. Introduction & Literature Review

1.1 Background

Computer simulations of real-world systems often times are very large, time consuming models that are analytically intractable with regard to identifying decisions that yield an optimal objective function value, due to the complex logical relationships that are internalized into the simulation. Even in deterministic simulations, it is necessary to run many replications just to find an approximate globally optimal solution. The primary reason for this inefficiency is because many of the traditional optimization techniques commonly used in linear and nonlinear environments cannot be applied directly to the output of the simulations, which do not yield a closed-form analytic equation for the objective function. Meta-models can overcome this problem by entirely replacing the simulation results with a mathematical function that approximates the output of the simulation.

Meta-models have been popularly defined as "models of a model" (see Badiru [7]) and are proven to be an effective analytic surrogate for extensive replications of a simulation, because classical optimization techniques can be applied directly to the functional form of a meta-model. If it is assumed that a meta-model has been verified to accurately estimate the simulation, the coordinates for the optimal solution of the meta-model will provide a set of simulation design points that can be tested directly in the simulation. Applied iteratively, this process of augmenting a

simulation with the functional form derived from a meta-model can reduce the total number of simulation replications that would have been needed in the absence of the meta-model. As with any effort to identify optimal solutions, regardless of the method invoked, improving the accuracy of the approximated solution is a critical point of interest for any researcher. Within this context, the improvement of meta-models is a significant part of the overall research objective in this dissertation.

Least squares regression is the most popular method to generate the functional form of a meta-model, when compared with more sophisticated techniques like neural networks and kriging, for three primary reasons: it is the most straightforward approach, it is easy to identify the most salient features, and it is better at inferring relationships between input and output variables. Despite these desirable properties, Lin et al. [72] find least squares regression to be sensitive to any model misspecification and to conform to any assumptions regarding the distribution of error, which is commonly Gaussian in nature. Myers et al. [84] indicate one of the only effective means to globally optimize a complex nonlinear simulation to be via regression, i.e., by sequentially optimizing small localized regions that are estimated using regression functions until a number of metrics have converged. However, this method does not guarantee the identification of a globally optimized solution. The inefficient nature of the process required by regression-based meta-models attenuates the aforementioned strengths of the regression method. In smaller applications, this method may prove useful; however, in practice on larger applications, its utility is diminished. Regression proves useful in cases where the underlying models can be approximated by linear regressors; however when the underlying model can only be estimated by non-linear effects, this method is often times abandoned and more state-of-the-art methods like neural networks or kriging are leveraged for very large simulations that have these non-linear attributes. The departure from least squares regression may be premature

in some applications as there may yet be improvements to identify for this method (e.g., largely with higher order interactions of control variates) beyond what has been published in the literature. These regression improvements will be a specific research focus of this work.

Despite the performance improvements that could be yielded from additional research into regression-based meta-models, in many cases regression will still be outperformed with regards to predictive accuracy by the most technological advanced alternative methods. One of the state-of-the-art methods is artificial neural networks, a set of powerful tools that have gained attention for their versatility. They can model, under certain conditions, any relationship to any degree of accuracy (given that enough nodes are introduced in the net framework) and, unlike regression based meta-models, Bao et al. [8] states these methods are insensitive to problems with model misspecification, assumptions of heteroscedascity or distribution of errors, multicollinearity and autocorrelation within the data. As mentioned earlier, neural network meta-models for simulation can address on the nonlinear environments that regression meta-models can struggle with while compensating for the difficulties that analysts have to consider. With its relative novelty, there seem to be many fundamental questions that have been left unanswered with the application of neural networks in simulation meta-models. Neural networks still have critics (e.g., see Myers et al. [84]) who view their role only as supplemental tools to regression because it is difficult to directly infer information about the relationship between its inputs and outputs. These questions amount to a significant portion of the specific research focus that are declared in the next section and which a future chapter elaborates.

1.2 Statistical Framework

In this section, we provide the basic groundwork that is used to underpin the extensions made in Chapters II through IV. This review is partitioned into three main components that cover control variates, neural networks, and simulation meta-models. The main sections are followed by three focused topics that are specific to the research in the subsequent chapters, that review Jackson network formulas for open networks, interior point optimization, and a brief summary of the simulation that is used throughout the research in this dissertation.

1.2.1 Control Variates (CV).

During the process of building stochastic computer simulations, parameters for stochastic processes that may exist in the internal logic of the model are programmed and therefore *known*. The cumulative effect of these stochastic processes and their interactions with themselves and all other components of the simulation provides the system dynamics that govern the random output(s) of a simulation, for which the statistical parameters are generally *unknown*. CVs are used as part of a strategy that exploit its correlation with the response of interest to reduce variance. These types of constructions, that will be referred here as control variates, have also been referred to as regression sampling as is done with Kleijnen et al. [58], or as concomitant variables in Ahmed et al. [1], Lavenberg et al. [67], and Wilson et al. [132, 131].

A progression from the simple univariate control with single response scenario to the more complex multivariate control variates with multiple simulation responses will be featured in the following sections. Unfortunately, despite the interesting work in non-linear control variate transformations by Nelson [87], Swain[117, 118], Glynn & Whitt [37], Szechtman [119], and Kim & Henderson [56], in the sections that follow, only linear transformations of the control variates are featured.

Control variates in the 1960's were commonly studied under a broader field of variance reduction techniques (VRTs) with other subtopics like antithetic variates, stratified sampling, selective sampling, among others; as is done with the survey by Hammersley & Handscomb [41] on Monte Carlo methods. However, the research on control variates was primarily centered on the analysis of the integration of Monte Carlo estimations. It wasn't until 1974 in Kleijnen's seminal two volume series [58], that a robust survey of VRTs (particularly Chapter III on control variates) primarily centered around the statistical analysis of discrete event simulations. Kleijnen distinguished the two key differences of Monte Carlo VRTs and simulation VRTs by their dependency. The observations in Monte Carlo studies are independent, while the observations in simulation are commonly dependent. The second division is that Monte Carlo can be characterized by a "rather simple function", while the simulation response(s) are often times analytically intractable and can only be expressed by "the computer program itself." Kleijnen's effort was followed by Wilson & Rubinstein in 1984 with two comprehensive VRT reviews [130] and [129] where he introduces a VRT taxonomy to better classify the techniques. Wilson classified "... all VRTs into two major categories – correlation methods and importance methods" and prescribes more detail to both of the methods. Under the correlation method he describes common random numbers, antithetic variate, and control variates; and places importance sampling, conditional Monte Carlo, stratified sampling, and systemic sampling under the importance method division. When these two broad categories are contrasted the fundamental support of the methods are the key distinction. Correlation methods harness the existing correlation in a simulation's variates to reduce variance, while the importance methods use a priori information of the domain to reduce variance. It should be noted here that Handscomb [43] says a method is variance reducing only, "if it reduced the variance proportionately more than it increases the work involved."

Similarly to Wilson, Nelson & Schmeiser [89] provided several surveys that most notably posed a revised VRT taxonomy, also see Nelson [86]. A common underlying theme that can easily be observed from these VRT taxonomies is the internalization of correlation to induce variance reduction. Furthermore, a distinct feature that control variates has over antithetic variates and common random numbers is that it can use both signs of correlation to induce variance reduction whereas the other methods can only use one sign. Lavenberg and Welch [69] said of all the techniques, "... the method of control variates is one of the most promising." Finally, control variates has become popular with simulation practitioners because the information pre-exists in the simulation, markedly in stochastic environments. Because of its utility, control variate theory has expanded quickly and broadly throughout the 1980s and 1990s, with the discoveries of entirely new types of control variates like Bauer's dissertation [14] in 1987 on standardized routing variables. It is precisely this reason for which Nelson [88] extended his original VRT taxonomy to include a five category control variate specific taxonomy in 1990, to better handle the many types of control variates with unique parameters that had been developed by this time.

Since the turn of the century, a noticeable decline in the development of new control variate theory has occurred. A large proportion of 21st Century publications are devoted to demonstrating the application of established control variate theory in various inter-disciplinary domains, considerably in the highly volatile fields of financial engineering.

Table 1 below is an extension to a table found in the literature survey of Meents' masters thesis [77], that categorizes prior control variate research in a logical and concise manner. The first category of columns under 'Control' identifies if the publication is based on single controls ('S') or multiple controls within a simulation ('M'). The second category of columns under 'Response' identifies if the publication is based

on single response ('S') or multiple response simulations ('M'). The third category of columns under 'Population' identifies if the publication is based on a single population of design points ('S') or multiple populations of design points ('M'). The fourth category of columns under 'Design' identifies if the publication is based on a single designed experiment ('S') or multiple designed experiments ('M'). The fifth category of columns under 'Variance' identifies if the publication assumes if the variance of the simulation is known ('K') or unknown ('U').

Table 1. Taxonomy of Control Variate Publications

	Control		Response		Population		Design		Variance	
Source	S	M	S	M	S	M	S	M	K	U
Nozari [92]		X	X			X	X		X	X
Cheng [25]		X	X		X		X		X	
Kleijnen [58]	X		X		X	X	X			X
Lavenberg [67]		X	X		X		X			X
Lavenberg [69]		X	X		X		X			X
Rubinstein [109]		X		X	X		X			X
Porta Nova [101]		X		X		X	X			X

From the information that can be gleaned from Table 1 above, it is apparent there are several cases that exist that deserve specific treatment. To maintain the flow of this literature review, the cases that pertain to multiple populations (or in other terms, multiple design points) align more with the theoretical contributions that are provided by meta-models, and as such, can be found under the meta-model section. Any cases that deal specifically, with a single population (or in other terms, a single design point) are treated below in this section because these cases have theoretical contributions that deal with more pure control variate theory.

1.2.1.1 Single Control Variate (CV) for Single Population, Single Response Simulation.

Lets assume a parameter θ has an unbiased estimate called y. This assumes then that $E[y] = \theta$. Now, let c be a random variable that is observed from a simulation and has a known expectation μ. This random variable c, is correlated with y by $-1 \leq \rho \leq 1$ and is called a control variate. Then there exists a constant b coefficient that gives the "controlled estimator" (see Bauer & Wilson [14]), $y(b)$, expressed by

$$y(b) = y - b(c - \mu_c). \tag{1.2.1.1}$$

From this formulation of a linear control variate in equation (1.2.1.1), the variance of the output from the simulation response y can be adjusted by the product of a constant b and the correlated control variate $(\bar{c} - \mu_c)$. This new output $y(b)$ has the same expected value as the original simulation output y

$$E[y(b)] = \mu_y, \tag{1.2.1.2}$$

see equation (B.1.0.1) of Appendix B for derivation. Now when the expected value function is passed through the linear transformation of the response y, the control variate terms fall out leaving only the original expected value of the response. This ultimately means this transformed response is an unbiased estimate of the original response of the simulation. The variance reduction of the response is controlled with the b coefficient in equation (1.2.1.1). Law [70] states this variance reduction of $y(b)$ from the original response y is achieved only when equation (1.2.1.3) holds true

$$2b \cdot \text{cov}(y, c) > b^2 \cdot \text{var}(c), \tag{1.2.1.3}$$

see (B.1.0.8) in Appendix B for derivation. Kleijnen [58] states an optimal value b^* can be calculated that will maximize the variance reduction of the adjusted response $y(b)$ by differentiating equation (1.2.1.3) and evaluating it at 0, giving

$$\frac{df}{db} = b^* = \frac{\text{cov}[y, c]}{\text{var}[c]}, \tag{1.2.1.4}$$

see (B.1.0.12) in Appendix B for derivation. Re-evaluating at b^*, gives

$$\text{var}[y(b^*)] = (1 - \rho_{yc}^2) \cdot \text{var}[y], \tag{1.2.1.5}$$

see equation (B.1.0.16) Appendix B for derivation. This asserts that as the correlation between the control variate c and the simulation response y strengthens to 1 or -1, the variance reduction will increase on the linearly transformed simulation response $y(b)$. In addition, it is apparent that the variance of the transformed response $y(b)$ can never have more variability than the original response y.

The variance reduction that is achieved is under the assumption that the optimum coefficient b^* is known. However in practice b^* is unknown, hence it must be estimated. This process of estimating b^* must be done in a way as to realize some degree of efficiency. Lavenberg et al. [69] says that the practical value of control variates is not assessed by the variance reduction that is attained. It is only after "the estimation of b^* and the incorporation of this estimate into the statistical procedure of interest" that efficiency is marked. Lavenberg et al. [69] says this efficiency is most commonly done through the calculation of the point estimate and its associated confidence interval.

To generate the confidence interval, a calculation of variance is required and the simplest method is with multiple control variate samples derived from independent replications of the simulation.

Following Lavenberg et al. [69], let μ be the unknown parameter of interest. The simulation can generate an unbiased point estimator of μ called y from the average of the uncontrolled observations. The variance, $\sigma^2[y]$ is estimated from N statistically independent and identical simulation runs. If y_i, $i = 1, \ldots, N$ be the point estimator for the i^{th} replication, the the unbiased estimator $\bar{y}$ is expressed as

$$\bar{y}(b^*) = \frac{\sum_{i=1}^{N} y_i(b^*)}{N}, \tag{1.2.1.6}$$

where the controlled response y_i is expressed as

$$y_i(b^*) = y_i - b^*(c_i - \mu_c), \quad i = 1, \ldots, N. \tag{1.2.1.7}$$

Now in practice, $\text{cov}[y, c]$ and $\text{var}[c]$ that are found on the RHS of equation (1.2.1.4), are unknown which makes the optimal control coefficient b^*, on the LHS of equation (1.2.1.4), unknown. Porta Nova [101] provides an approach that replaces the RHS of equation (1.2.1.4) with their associated sample statistics, which yields the least squares solution. Bauer [14] notes that under joint normality assumption between y and c the "least squares solution is also the maximum likelihood solution." Kleijnen [58], gives more detail of the proof of this derivation. Treating the estimation in several methods, including splitting observations or using jackknife method, that all arrive at this same conclusion.

See equation (B.1.0.22) in Appendix B for regression theory of the least squares estimate for a formulation of the linear regression model which can derive the parameter estimates for μ_y and b^*.

1.2.1.2 Multiple Control Variates for Single Population, Single Response Simulation.

A scalar CV, $\bar{c}_i$, is defined as the observed average across $j = 1, \ldots, N$ instances from the i-th stochastic process inside a simulation, and given as

$$\bar{c}_i = \sum_{j=1}^{N} \frac{c_{ij}}{N}, \tag{1.2.1.8}$$

A $Q \times 1$ column vector of controls can be denoted by $C = [\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_Q]'$, where each scalar CV $\bar{c}_i$ has an associated *known* true mean μ_i. Now let $\mu_C = [\mu_1, \mu_2, \ldots, \mu_Q]'$ be a $Q \times 1$ column vector of those known means for each respective control in C. Further, let $B = [b_1, b_2, \ldots, b_Q]$ be a $1 \times Q$ row vector of constants. The controlled estimate, $y(B)$ for the response y is then given as,

$$y(B) = y - B'(C - \mu_C). \tag{1.2.1.9}$$

We can see that this provides an unbiased estimate for y because $E\{C\} = \mu_C$, giving us $E\{y(B)\} = E\{y\}$. If we let the covariances between the CVs and dependent response variable, $\text{var}\{y, C\}$, be given as a $1 \times Q$ row vector, and let $\Sigma\{C\}$ be a $Q \times Q$ covariance matrix amongst the CVs, then the optimal variance reducing vector of coefficients, β, can be calculated as

$$\beta = \text{cov}\{y, C\} \cdot \Sigma^{-1}\{C\}. \tag{1.2.1.10}$$

When β is substituted into equation (1.2.3.31) for B, and after some basic calculus and algebraic steps, this will yield what Lavenberg, Moeller, & Welch [68] call the "minimum variance ratio",

$$\text{cov}\{y(\beta)\} = (1 - R^2\{y, C\}) \cdot \text{var}\{y\}, \tag{1.2.1.11}$$

where $R^2\{y, C\}$ is the coefficient of determination (or squared coefficient of multiple correlation). It is important to distinguish that equation (1.2.3.33) assumes that β from equation (1.2.3.32) is *known*, when in most cases this information is generally *unknown* and therefore must be estimated. The sample analog for equation (1.2.3.32) is then given as

$$\hat{\beta} = S\{y, C\} \cdot S^{-1}\{C\}, \tag{1.2.1.12}$$

where

$$S\{y, C\} = \frac{\sum_{j=1}^{N}(y_i - \bar{y})(C_j - \overline{C})'}{N - 1}, \tag{1.2.1.13}$$

and

$$S\{C\} = \frac{\sum_{j=1}^{N}(C_j - \overline{C})(C_j - \overline{C})}{N-1}. \tag{1.2.1.14}$$

The vector of constants, B, in the controlled estimate of y from equation (1.2.3.31) is now substituted with $\hat{\beta}$,

$$\bar{y}(\hat{\beta}) = \bar{y} - \hat{\beta}'(\overline{C} - \mu_C). \tag{1.2.1.15}$$

where $\overline{C}$ is a $Q \times 1$ column vector of the sampled means of the Q CVs.

1.2.1.3 Multiple Control Variates for Single Population, Multiple Response Simulation.

Rubinstein [109] extended the work of Lavenberg [68] from the previous section, by generalizing the formulation of equation (1.2.1.7) to estimate multivariate mean responses with multivariate controls.

Equation (1.2.1.7) is now given as

$$Y(B) = Y - B(C - \mu_C), \tag{1.2.1.16}$$

where Y is a $P \times 1$ column vector of responses given as

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{bmatrix}, \tag{1.2.1.17}$$

$Y(B)$ is a $P \times 1$ column vector of controlled responses given as

$$Y(B) = \begin{bmatrix} y_1(B) \\ y_2(B) \\ \vdots \\ y_P(B) \end{bmatrix}, \tag{1.2.1.18}$$

B is a $P \times Q$ matrix of control coefficients given as

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1Q} \\ b_{21} & b_{22} & \cdots & b_{2Q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{P1} & b_{P2} & \cdots & b_{PQ} \end{bmatrix}, \tag{1.2.1.19}$$

C is a $Q \times 1$ column vector of controls, and μ_C is a $Q \times 1$ column vector of of known means.

Similarly to the two previous sections the variance of $Y(B)$ is minimized by a generalized formulation given as

$$B^* = \Sigma_{YC}\Sigma_{CC}^{-1}, \tag{1.2.1.20}$$

where the generalized form of the covariance between the controls and the responses is now given as

$$\Sigma_{YC} = E[(Y - \mu_Y)(C - \mu_C)'], \tag{1.2.1.21}$$

and the generalized form of the covariance between the controls is now given as

$$\Sigma_{CC} = E[(C - \mu_C)(C - \mu_C)]. \tag{1.2.1.22}$$

Bauer [12] reviews the generalized form of the minimum variance as

$$|\text{cov}[Y(\beta)]| = |\Sigma_{YY} - \Sigma_{YC}\Sigma_{CC}^{-1}\Sigma_{CY}| = |\Sigma_{YY}| \prod_{i=1}^{V}(1 - \rho_i^2), \tag{1.2.1.23}$$

where $V = \text{rank}(\Sigma_{YC})$ and $\rho_i^2, i = 1, \ldots, V$ are the canonical correlations between Y and C that satisfy $\rho_1 \geq \rho_2 \geq \cdots \geq \rho_V$.

Rubinstein [109] gives the proof for the $100(1-\alpha)\%$ confidence interval for μ_Y as

$$Pr(N(\bar{Y} - \mu_Y)'\Sigma_{YY}^{-1}(\bar{Y} - \mu_Y) \leq \chi_{p,1-\alpha}^2) = 1 - \alpha, \tag{1.2.1.24}$$

where

$$\bar{Y} = \frac{\sum_{i=1}^{N} Y_i}{N}. \tag{1.2.1.25}$$

The volume of this ellipsoid is given by

$$V_1 = p^{-1} C(p) |\Sigma_{YY}|^{1/2} (\chi^2_{p,1-\alpha}/N)^{p/2}, \tag{1.2.1.26}$$

where

$$C(p) = 2\pi^{p/2}/\Gamma(p/2). \tag{1.2.1.27}$$

1.2.1.4 Control Variate Selection.

Calculating the correlation and associated variance reduction obtained by the use of control variates is only one part of the overall variance reduction strategy. The second piece of the process addresses the question of which control variates are the best to include or discard. The most obvious and intuitive objective is to choose the control variates that are the "most strongly correlated with the output random variable ... and would also like the control variates themselves to have low variance." [70]

There have been many papers that have provided criteria or efficiency measures for control variate selection. Lavenberg et al. [68] provided dual approaches that use either all available controls or including only the three best control variates using stepwise regression. They provided two efficiency metrics: the minimum variance ratio and the loss factor. Rubinstein & Marcus [109] provided criteria for calculating

the variance reduction loss that comes with estimating the optimal control matrix for both univariate and multivariate scenarios. Venkatraman & Wilson [122] extend the findings of Rubinstein & Marcus with a new variance ratio and loss factor that is computationally more simple. Porta Nova & Wilson [101] provided a generalized framework for the previous publications as well as extending the application of control variates to multiple response simulation meta-models. Bauer & Wilson [15] developed a procedure, that under several assumptions about the variance and covariance of control variates, select the best subset from the total set of available control variates. Porta Nova & Wilson [102] extends upon their prior work in selecting an optimal subset of control variates under the context of multiple response simulation meta-models.

Law [70] describes three sources of control variates: internal, external, and convex combinations of the estimators.

1.2.1.5 Internal Control Variates.

Internal control variates as described by Law [70] are variates that can be observed with very little computational expense to the overall variance reduction endeavor. The parameters for these internal variates are generally known, as they are programmed explicitly into the simulation. Even simple analysis of their relationships with other parts of the simulation could suggest that they might be correlated in one direction or the other with the random output. A more detailed accounting of these types of control variates are found in Iglehart & Lewis [48], Lavenberg et al. [67, 68], and Wilson [132].

1.2.1.6 External Control Variates.

This second method calls for using common random numbers in a second analytically tractable simulation that is similar to the original simulation. The output of this second simulation is treated as control variates for the original system, hence being named external control variates. Intuition would lead a practitioner to believe that this second system is somehow significantly correlated with the output of the first simulation by utilizing common random numbers. These types of control variates though require more computational expenses in the form of more simulation runs.

1.2.1.7 Combination of Control Variates.

In many scenarios, there are several estimators to choose from, and as such, these variates can be combined into a single estimator. Using Law's [70] notation, let $c_1, \ldots, c_Q$ be a series of length Q that are all unbiased estimators for μ, but not necessarily independent of one another. If $b_1, \ldots, b_Q$ be a series of any real number that sum to 1, then

$$c = \sum_{i=1}^{Q} b_i c_i \tag{1.2.1.28}$$

is also unbiased for μ. Now intuitively, since $b_1 = 1 - \sum_{i=2}^{Q} b_i$, then c can be expressed as

$$c = c_1 - \sum_{i=2}^{Q} b_i (c_1 - c_i), \tag{1.2.1.29}$$

see equation (B.1.0.48) in Appendix B for derivation. Now the combination of the c_i's can be viewed as $Q - 1$ control variates for c_1 if we let $y_i = c_1 - c_i$, for $i = 1, \ldots, Q$.

Now if the simulation model is very large with many stochastic processes, then the number of internal, external, and combinations thereof will expand very quickly, so the selection of the "best" subset of control variates can become a very difficult task.

1.2.1.8 Standardized "Work" Control Variates (CVs).

By standardizing CVs, difficulties that result from disparate units of measurement can be avoided. Wilson & Pritsker [132] notes that un-standardized CVs are unstable for increasingly long run-lengths and therefore defined the standardized "work" CV, an asymptotically stable CV as

$$C_i(t) = [a(i,t)]^{-1/2} \sum_{j=1}^{a(i,t)} \frac{U_j(i) - \mu_i}{\sigma_i} \tag{1.2.1.30}$$

where $U_j(k) : j \geq 1, i >= 1$ is an IID sequence of the observed jth instances from the ith stochastic process in the period $[0,t]$, μ_i is the true *known* mean of the ith stochastic process, and $a(i,t)$ is the total number of samples observed. Wilson & Pristker show if the regenerative property holds, than the $Q \times 1$ column vector $C(t) = [C_1(t), \ldots, C_q(t)]'$ of standardized CVs will converge as,

$$C(t) \rightarrow N_Q(0_Q, I_Q) \text{ as } t \rightarrow \infty. \tag{1.2.1.31}$$

Also see Wilson & Pritsker [132] for an embellishment to Lavenberg, Moeller, & Welch [68], that standardizes their "work" variables for closed queueing networks, making them stable.

1.2.1.9 Standardized "Routing" Control Variates (CVs).

Lavenberg, Moeller, & Welch [68] first treated stochastic multinomial branching processes under the context of a closed queueing network, also generally common in simulations, as "flow" CVs. Bauer & Wilson [16] introduced the idea of "routing" CVs as an asymptotically standardized version of these "flow" CVs. This new category of CVs are standardized for similar reasons as "work" CVs that are covered in Subsection 1.2.1.8. The indicator function for this multinomial branching process is given as,

$$I_j(g) = \begin{cases} 1 & \text{if the } j\text{th departing request goes to activity } g \\ 0 & \text{otherwise} \end{cases} \tag{1.2.1.32}$$

where g branches direct the flow of activity from one process to g subsequent processes. The standardized "routing" CV for this multinomial branching process is then given as

$$R_g = \sum_{j=1}^{N(t)} \frac{I_j(g) - p(g)}{\left[N(t)\left[1 - p(g)\right] p(g)\right]^{1/2}}, \quad g = 1, \ldots, G, \tag{1.2.1.33}$$

where $N(t) > 0$ is the total number of transits through all G branches in the time interval $[0, t]$, $p(g)$ is the known true probability for the gth branch, and $I_j(g)$ is the indicator function for the observed sampling of that gth branch. Bauer & Wilson [16] show the $G \times 1$ column vector $R = [R_1, \ldots, R_G]'$ of standardized "routing" CV will converge to a joint normal distribution with a null mean and nonsingular covariance matrix Σ_R given as

$$R \xrightarrow{x} N_g(0, \Sigma_R) \text{ as } t \to \infty. \tag{1.2.1.34}$$

where

$$(\Sigma_R)_{jk} = \begin{cases} 1 & \text{for } j = k \\ -\left\{ \dfrac{p(j)p(k)}{[1-p(j)][1-p(k)]} \right\}^{1/2} & \text{for } j \neq k. \end{cases} \tag{1.2.1.35}$$

See Appendix 4, pg 141 of Bauer [12] for the derivation of equations (1.2.1.34) and (1.2.1.35).

1.2.1.10 Method of Replication with Control Variates (CVs).

The method of replication with CVs provides an unbiased estimate for μ_y. When under favorable conditions this method will give a smaller variance then the estimate provided by the method of direct simulation. To construct a confidence interval for μ_y, we first assume the distribution of y and C from equation (1.2.1.36) is jointly normal [68],

$$\begin{bmatrix} y \\ C \end{bmatrix} \sim M_{Q+1} \left[\begin{bmatrix} \mu_y \\ \mu_C \end{bmatrix}, \begin{bmatrix} \text{var}\{y\} & \text{cov}\{y, C\} \\ \text{cov}\{C, y\} & \text{var}\{C\} \end{bmatrix} \right], \tag{1.2.1.36}$$

Then we can evaluate $\bar{y}(\hat{\beta})$ from equation (1.2.1.37), as an unbiased point estimator for μ_y, giving us a $100(1-\alpha/2)\%$ confidence interval,

$$\bar{y}(\hat{\beta}) \pm t_{1-\alpha/2,M-Q-1} \cdot D \cdot S\{y, C\}, \tag{1.2.1.37}$$

where

$$S\{y, C\} = \sqrt{\frac{M-1}{M-Q-1} \cdot [\hat{\text{var}}\{y\} - S\{y, C\} \cdot S^{-1}\{C\} \cdot S'\{y, C\}]}, \tag{1.2.1.38}$$

and

$$D = \sqrt{\frac{1}{M} + \frac{(\overline{C} - \mu_C)' \cdot S^{-1}\{C\} \cdot (\overline{C} - \mu_C)}{M-1}}, \tag{1.2.1.39}$$

and $t_{1-\alpha/2,M-Q-1}$ is the $100(1-\alpha/2)\%$ quantile of the Student's t-distribution with $M-Q-1$ degrees of freedom.

1.2.1.11 Method of Direct Simulation.

In the absence of variance reducing CVs, the method of direct simulation is the traditional formulation for constructing a $100(1-\alpha/2)\%$ confidence interval for the mean of an arbitrary response of interest, μ_y. If there exists M independent replications of a simulation then an unbiased estimate for μ_y is given by

$$\hat{\mu}_y = \bar{y} = \sum_{i=1}^{M} \frac{y_i}{M}. \tag{1.2.1.40}$$

If we reasonably assume y_i are independent and identically distributed (IID), and normally distributed, evaluating the product of the $100(1-\alpha/2)\%$ percentile of a

t-distributed random variable (with $M-1$ degrees of freedom), with the square root of the predicted variance of the mean $\bar{y}$

$$\hat{\text{var}}\{\bar{y}\} = \frac{\hat{\text{var}}\{y\}}{M}, \tag{1.2.1.41}$$

where

$$\hat{\text{var}}\{y\} = \frac{\sum_{i=1}^{M}(y_i - \bar{y})^2}{M-1}, \tag{1.2.1.42}$$

will give us the necessary pieces to formulate a $100(1-\alpha/2)\%$ confidence interval for μ_y, given by

$$\hat{\mu}_y \pm t_{M-1,1-\alpha/2} \cdot \sqrt{\hat{\text{var}}\{\bar{y}\}}. \tag{1.2.1.43}$$

1.2.2 Neural Networks.

Artificial neural networks (ANN) are inspired by the natural cognitive systems, as illustrated in Figure 1. ANNs are constructed as networks of simple computational units that work together simultaneously. The nature of the connections between the computational units allow a form of communication that passes information from one node to another. As this information is funneled to a computational unit, the simple mechanism that is programmed into the node will process the information that arrives and send forward the transformed signal to the next connected node. All of this processed information flows simultaneously, and in parallel, through the network of nodes.

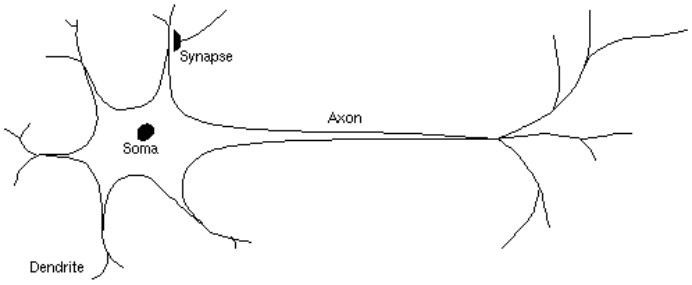


Figure 1. Sketch of biological neuron. [50]

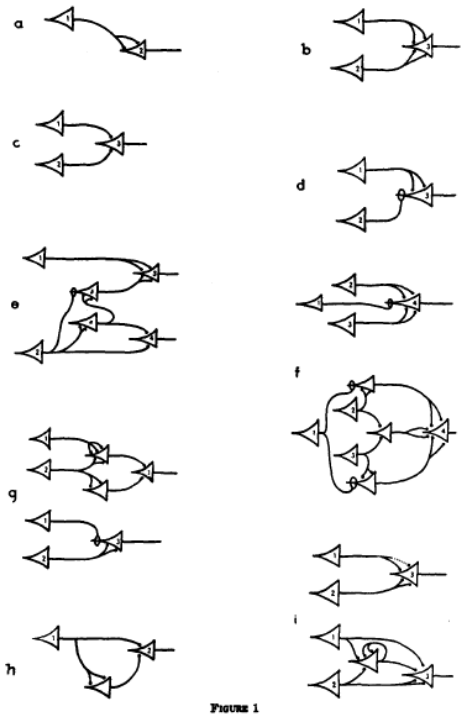


Figure 2. McCulloch illustrated network of nodes. [76]

In 1943, McCulloch et al. [76] first introduced the basic principle of an interconnected map of nodes that used discontinuous step functions, as is illustrated in Figure 2, modeling their behavior with a mathematical proof that shows, "any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes." This work by McCulloch, however, lacked the ability to digitally "learn".

Using the material from McCulloch et al. [76] and Jain's notation [50], the proposal by McCulloch gave a step function as the mathematical model for the individual node of a neural network. This node computed a weighted sum of its input signals, $x_j, \quad j = 1, \ldots, n$ that generates a binary output. This node will output 1 if the sum of input signals exceed a threshold u, otherwise 0 is the outputted result. In Duda's text [29], the activation functions replace the output of 0 with -1.

$$y = \theta(\sum_{j=1}^{n} w_j x_j - u), \tag{1.2.2.1}$$

where the step function is represented by $\theta(\cdot)$ and the j^{th} weight and threshold that is associated with the perceptron, as w_j and u respectively. This mathematical model is illustrated in Figure 3 below.

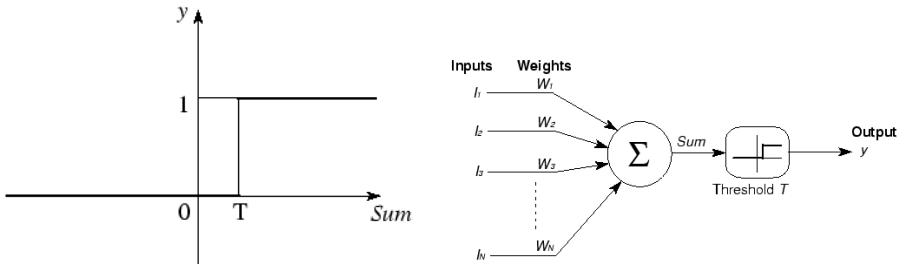


Figure 3. Linear Threshold Function (Left) and McCulloch & Pitts model (Right) [76] [50]

The second major neural net breakthrough addressed the inability of McCulloch's neural network to "learn". In the late 1950s and early 1960s through several works by Frank Rosenblatt [106, 108, 107], who recognized and provided the first proofs that show a directed network of nodes would need to adapt and evolve by iteratively updating itself, until it converges. In Rosenblatt's learning algorithm, the first step and second step is to initialize the k weights and threshold randomly after which a vector $(x_1, \ldots, x_n)^i$ is evaluated. The final step evaluates the function below,

$$w_k(i+1) = w_k(i) + \pi(c-y)x_k, \tag{1.2.2.2}$$

where c is the desired output, y is the actual output, i is the iteration, and π is a step size multiplier that has a range from 0 to 1.

Throughout the 1950s and 1960s, quite a bit of turmoil amongst the research community centered on the general utility of these perceptron neural networks, which is well documented by Olazaran [94] and Widrow [127]. They document the critics like Minsky and Papert [78], who were critical to Rosenblatt's findings, that did not like the "mystique" surrounding "loosely organized neural network machines" who concluded that no progress could be made with neural networks and all efforts should be abandoned. However, as described by Olazaran [94], other contributors with a more positive view of neural nets, like Widrow and Rosen, found Minsky and Papert's position as "irrelevant". Up until 1986, the functions used in a neural network were perceptrons (right side of Figure 3) that commonly used non-differentiable linear threshold (step) functions (left side of Figure 3). The third major breakthrough in neural network theory is in the development of the six step back-propagation of error algorithm, which is based on Rosenblatt's original three step error-correction principle.

This learning mechanism, unbeknownst to most of the research community for many years was first proposed in Werbos' Harvard PhD thesis [125] in 1974. In 1986, Rumelhart et al. [110] brought the neural network back-propagation method to the mainstream primarily due to the augmentation of the perceptrons with continuous non-linear functions called sigmoids. This was a major breakthrough as this reformulation of the method made it possible to generalize all prior work to multilayered networks [94]. Gaussian and piecewise linear functions have since been introduced as

additional activation functions beyond threshold functions and sigmoids, see Figure 4.

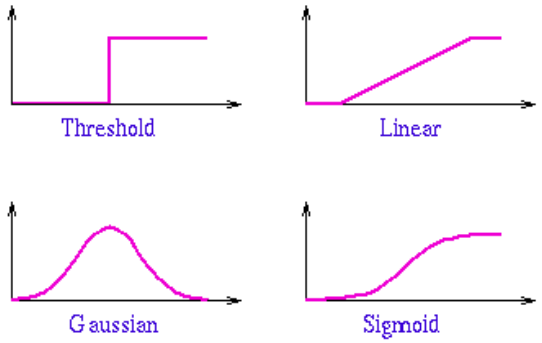


Figure 4. Types of activation functions. [50]

From Rumelhart's contribution, "An enormous number of papers and books of applications, from speech production and perception, optical character recognition, data mining, finance, game playing and much more, continues unabated." [29] Over the past 30 years an entire taxonomy of neural networks have been developed that is best illustrated by Jain's publication [50] in Figure 5.

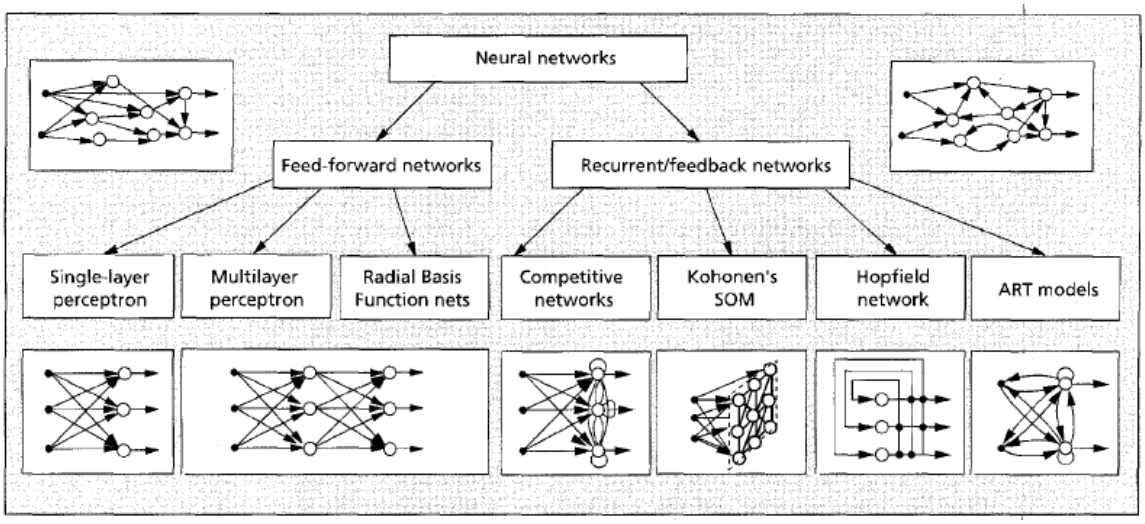


Figure 5. Taxonomy of neural net architectures. [50]

The content of this section up to this point, has contributed towards a general understanding of the steps taken to formulate the conventional feed-forward multilayer neural network that uses the back-propagation algorithm to learn. For the purposes of this research, the remaining content is limited to Radial Basis Neural Networks (RBNNs), a specialized formulation of the feed-forward construct. See Duda et al. [29] for comprehensive analysis that broadly covers all types of ANNs. Also see Section B.2 of Appendix B for review of artificial feed-forward neural networks.

1.2.2.1 Fundamentals of Radial Basis Neural Network (RBNN).

Radial basis neural networks (RBNNs) fall under a sub-category of ANNs called probabilistic neural networks (PNNs), that were first proposed by Specht [115]. The primary characteristic for these types of ANNs is that the network substitutes the sigmoid function with a radial basis function. Evaluation on the distance between an input pattern and the specific pattern or center associated with each middle node (see Figure 6).

Figure 6 illustrates the RBNN with an n dimensional vector of inputs, X, an array of m exponential basis functions, H, a vector of m weights for each of the basis functions, W, and the single node Σ for the output, Y, with bias, B.

More specifically for the hidden layer in Figure 6, the exponential function basis neuron, H_i, is defined by the vector of $m \times 1$ column vector of exponential functions, given as

$$H_i(X) = \exp\left(\frac{-D_i^2}{2\sigma^2}\right), \tag{1.2.2.3}$$

where

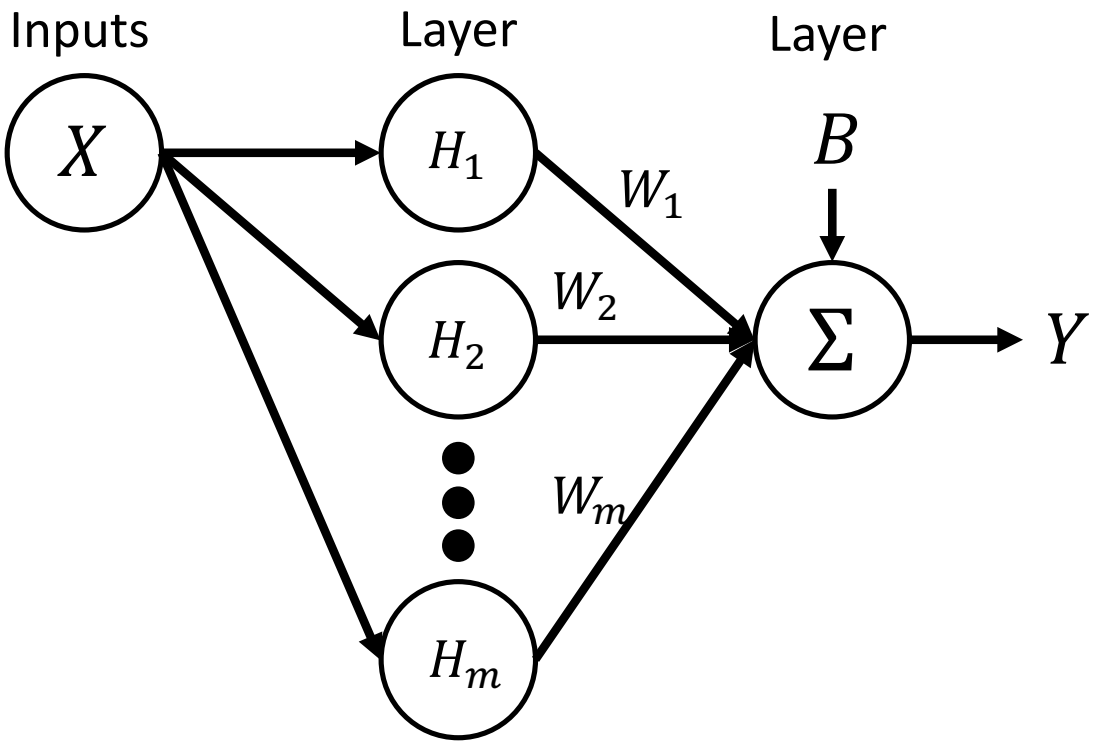


Figure 6. Example Radial Basis Neural Network (RBNN) *excluding* Standardized "Work" and "Routing" Control Variates

$$D_i^2 = (X - \mu_i)^T (X - \mu_i), \tag{1.2.2.4}$$

and X is an $n \times 1$ column vector of designed input variables, μ_i is an $m \times 1$ column vector of weights associated with the $i = 1, 2, \ldots, m$ exponential basis function neurons in the hidden layer, σ is a $m \times 1$ column vector of spreads that control how fast the neurons value decreases monotonically as the input, X, moves away from the center of the neuron, the bias as B, and the output Y is defined as

$$Y = \sum_i H_i \cdot W_i + B. \tag{1.2.2.5}$$

Wassermann [124], gives the gradient descent algorithm for the RBNN weights, W_i, that are trained iteratively by evaluating,

$$W_i(k+1) = W_i(k) + \eta \cdot (T - Y) \cdot H_i(X), \tag{1.2.2.6}$$

where W_i are the weights, k is the iteration ID, η is the step size, T represents the target response, Y is the predicted response, and $H_i(X)$ is the ith exponential function basis node. Specht [115] notes the training speed associated with RBNNs is much faster than the back propagation of the feed forward neural network.

Orr [95], gives the learning algorithm for the RBNN bias, B, trained by iteratively evaluating through steepest descent,

$$B(k+1) = B(k) + \eta \cdot (T - Y), \tag{1.2.2.7}$$

where B is the bias, k is the iteration ID, η is the step size, T represents the target response, and Y is the predicted response.

An ongoing topic in literature for RBNNs is the optimal selection of the number of neurons and their spread. Flietstra and Bauer [32] provide an approach for combining both feature and architecture selection for RBNNs through clustering and a signal-to-ratio (SNR), which is a direct consequence of the feature selection method based on SNR by Bauer et al. [13]. Their approach is similar to one taken for feedforward neural networks by Steppe, Bauer, and Rogers [116]. Wasserman [124] also provides a method for selecting the spread through a k nearest neighbor clustering approach.

1.2.2.2 Design of Experiments with Neural Nets.

A majority of issues that surround modeling is in regards to the reduction of variance. Entire fields of research based on designed experiments have been developed to address these issues. A lot of these methods were developed to compensate for the heteroscedastic sensitivity that exists for regression modeling techniques (see Myers [85]), which is the primary technique to develop representative models of the design space. Other modeling techniques, like splines, artificial neural networks, and many others, have been developed that entirely replaces regression as the means to an effective predictive model. Fishwick [31], Himmel and May [44], Han et al. [42], Hussain et al. [47], and Carpenter and Barthelemy [24] have suggested that artificial neural networks are particularly effective surrogates for regression based response surface methods because of its insensitive to heteroscedasticity, multicollinearity, and bias issues. In support, Padgett and Roppel [96] states,

> "Neural networks in all categories address the need for rapid computation, robustness, and adaptability. Neural models require fewer assumptions and less precise information about the systems modeled than do some more traditional techniques."

However contrary to this, Myers [85] has said that neural networks are no better than a

> "complement to the familiar statistical tools of regression analysis, RSM, and designed experiments, but certainly not a replacement for them. ... Furthermore, there is no reason to believe that the prediction properties of these models are superior to those that would be obtained from a well-designed RSM study."

In each of the sections that follow, a review of literature that examined the combination of the respective experimental design with an artificial neural network is

provided. Any and all conclusions on comparative studies across types of designs will be remarked on, albeit, there is not many beyond Alam et al. [2]. His research was the most comprehensive that specifically examined the effects of combining neural networks across several different designs with the intent of selecting a best design. Overall, the far majority of the work that has been done with the mixtures of experimental design and neural networks was on factorial, central composite, and latin hypercube design.

1.2.2.3 Factorial Design.

Of all the experimental designs, factorial are the most widely used when examining several factors. This design is useful for investigating the joint effects of the factors and their interactions on a response variable. Myers says [85] that factorial designs find applications in screening experiments, steepest ascent models, and when augmented with additional points become central composite designs (which is addressed later). Montgomery [79] says, "some disadvantages are the high number of experiments and that it does not consider possible curvatures." Alam et al. [2] support this notion, "the limited number of input levels involved in the factorial approach, the configurations are not capable of fully capturing the non-linear behavior of the response surface."

Now suppose that three factors, A, B, and C, each at two levels, are of interest. Then the design is called a 2^3 factorial design, and the eight treatment combinations can be displayed graphically as a cube, as is done in Figure 7.

Across the literature, it was found that Marengo et al. [74], Amato et al. [4], and Choudhury & Bartarya [26] each investigated the combination of a full factorial experimental design with a neural network using applications across varied industries. Alam et al. [2] showed the factorial design did not perform as well as other designs (mentioned later) when combined with a neural network.

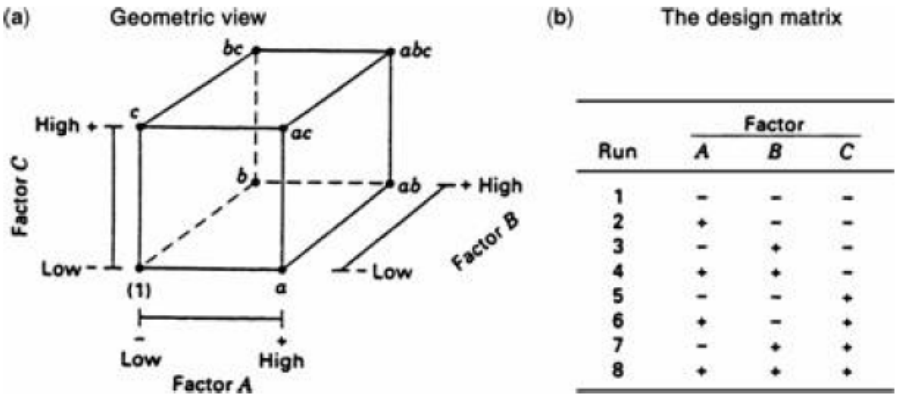


Figure 7. Example 2^3 factorial design [84]

1.2.2.4 Latin Hypercube Design.

The Latin Hypercube design, illustrated in Figure 8, is one of many types of "space-filling" designs. These designs will distribute design points across the design space so that they are uniformly scattered. Montgomery [79] says "there are a number of algorithms for creating these designs and several measures of uniformity." Now, space-filling designs have been considered be particularly suitable for computer models because they distribute the points across a space where the experimenter doesn't know the form of the model that is sufficient. Li et al. [71] worked with neural networks and Latin Hypercube designs noting,

> "Latin Hypercube designs are used as it provides good uniformity and coverage in the sample space, and is flexible on sample sizes. Latin hypercube designs treat all design variables with equal consideration and the range of each variable is stratified into m non-overlapping equally sized intervals."

Across the literature, it was found that Alam et al. [2], Lunani et al. [73], Jung & Yum [52], Shan et al. [113], and Kuo et al. [66] each investigated the combination of a latin hypercube design with a neural network using applications across varied

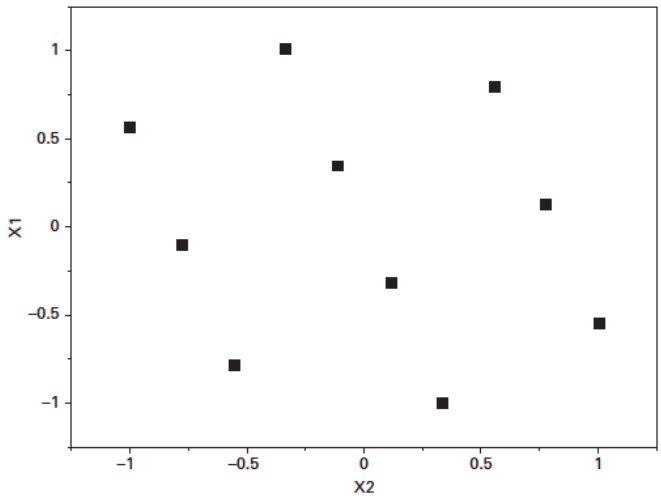


Figure 8. Example Latin Hypercube design with 10 points [80]

industries. Particularly important to note that Alam's [2] examination showed the latin hypercube design performed better than all other methods.

1.2.2.5 Central Composite Design.

Central composite designs, illustrated in Figure 9, are essentially a traditional factorial designs that have been augmented with additional center points and/or axial points. Montgomery [79] says, "the central composite design is one of the most important designs for fitting second-order response surface models." Two parameters that are important in this design are the number of center points and the distance of axial runs (which can be important in terms of a design being rotatable also known as constant variance).

Across the literature, it was found that Marengo et al. [74], Amato et al. [4], Novotna et al. [91], and Desai et al. [28] each investigated the combination of a central composite design with a neural network using applications across varied industries. Alam et al. [2] showed the central composite design did not perform as well as other designs when combined with a neural network. Also noting that the

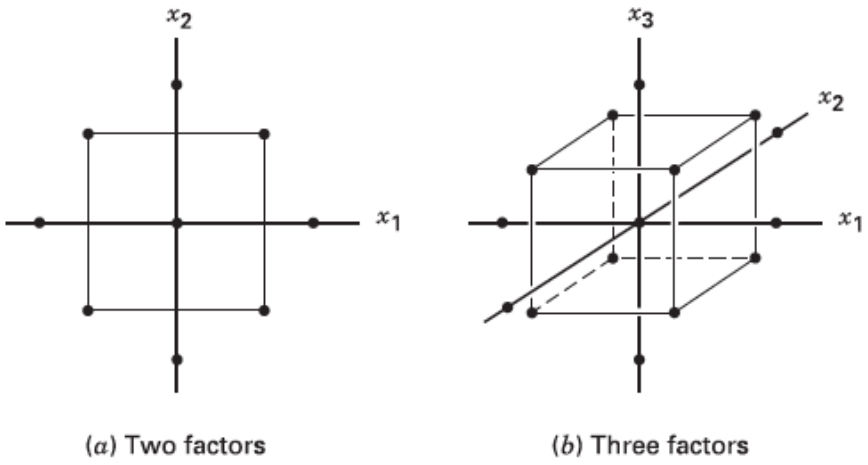


Figure 9. Example Central Composite Design with center and axial points [79]

"central composite design performs badly here is because the approach did not cover much of the design space while generating the training data set for the approach".

1.2.2.6 Other designs.

Anecdotally, other pieces of literature were uncovered that utilized other types of designs. Slokar et al. [114] used a Plackett-Burman design with a neural network. Haines [40] developed a framework for constructing D-optimal designs for neural networks which seemed to conclude with ambiguous results due to questionable assumptions on the underlying markov chain Monte Carlo metropolis algorithm. This was was followed by Issanchou & Gauchi [49] and Kim [23]. Finally Maran et al. [103] used a Box-Behnken experimental design in conjunction with a neural network.

1.2.3 Simulation Meta-models.

Simulations have been a long standing practice used to study and examine complex systems, that can quickly diverge in terms of computational, or even exorbitant monetary, expenses. The idea of capturing the outputs accurately of an expensive

simulation while avoiding the costly iterative nature of studying the simulation is where meta-modeling begins. Meta-models in the realm of operations research is defined as a method that construct functions that can approximate the outputs of large simulations that have as part of their internal structure, complex higher dimensional and non-linear relationships, that makes the simulation sophisticated, yet analytically intractable in some cases. Meta-modeling has many common roots from the well established and frequently researched field of response surface methodology. Much of the theory that exists from response surface methodology are transposed directly into methods that are characterized as meta-modeling practices.

Building models that approximate the output of a system begin in the 1950s with response surface methodology. One of the primary founding fathers and frequent contributor for this field of experimentation is G.E. Box, with his seminal work [20] suggesting the use of quadratic polynomial regression models in 1951. From his research, and many other contributions that came after, new techniques have expanded the field of response surface methodology significantly. Entirely new designs from composite designs, kriging, Plackett-Burman designs, latin hypercube, to various types of factorial designs. Each of these methods are uniquely efficient for particular designs able to handle varying types of problems, however, the use of artificial neural networks emerged in the 1990s that has proven to maintain high benchmarks more universally across a wide swath of problems, see Anjum et al. [30]

As simulations have become more advanced, complex, and computationally expensive, the common techniques used in response surface methodology was found to be useful in estimating the outputs of these simulations. Kriging (see Kleijnen [62]), radial basis functions (see Hussain et al. [47]), support vector regression (see Alenezi et al. [3]), and polynomial regression techniques (see Noguera and Watson [90]) have been tested in applications with meta-modeling. In a broader approach illustrated

in Figure 10 below, Li [71] highlighted where such types of meta-model could ideally occur in the chain of a decision support system.

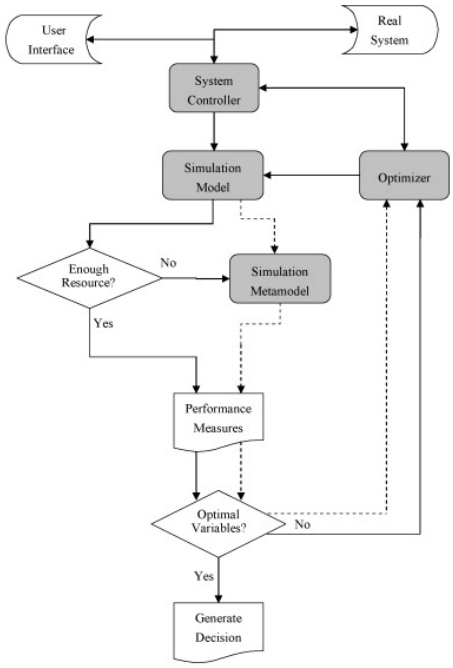


Figure 10. Simulation meta-model in a Decision Support System [71]

1.2.3.1 Meta-model Fundamentals.

Notation from Friedman [35] is followed here. An event that is replicated by a simulation is assumed to be visualized abstractly by equation (1.2.3.1) below,

$$\psi = \omega(\eta_1, \ldots, \eta_N), \tag{1.2.3.1}$$

where ψ is the particular event result of interest, ω is the characterization of the governing relationship between the N unknown number of factors that is represented as η_n, $1 \leq n \leq N$.

The simulation practitioner is interested in recreating computationally, much of the observable relationships from equation (1.2.3.1) above, from subject matter expert knowledge or through direct measurement. The simulation can be visualized abstractly as,

$$\psi \approx \gamma_i = g(z_{i1}, z_{i2}, \ldots, z_{im}) + \varepsilon_i \quad 1 \leq m \leq M, \tag{1.2.3.2}$$

where the real world event result ψ is approximated by the i^{th} iteration of the artificially simulated output γ_i, that is governed by the relationships constructed inside the simulation characterized by the function g, upon the M known number of internalized factors that is represented as z_{im}, $1 \leq m \leq M$. The deviation of the i^{th} simulated response γ from the true real-world result ψ is represented by $\psi - \gamma_i = \varepsilon_i$. It is important to note that the relationship between the z_{im} factors from equation (1.2.3.2) and the η_n factors from equation (1.2.3.1) depend on the event that is being simulated. There are many assumptions that will help evaluate the saliency of the relationship. In some cases, a single factor η_n may be exactly equivalent to z_n. It is almost a certainty that not all η_n will be incorporated into the simulation equation for various reasons due to measurement error or lack of information. And in other cases, there may be z_n that exist in the simulation that do not exist as η_n because the simulation practitioner is falsely assuming a factor contributes to the output, when in fact it does not. However, it can be assumed the simulation practitioner seeks to copy as many η_n into equivalent z_n formulations as possible.

As is mentioned in the introduction of this section, directly analyzing the simulation output γ_i of a very large, complex simulation is a difficult task, if not entirely impossible. The two levels of abstraction illustrated above is not enough to arrive to a state of output that can be analyzed. This means a third level of abstraction,

via a meta-model, is needed to arrive to this point. The meta-model is abstractly visualized in equation (1.2.3.3) as

$$\psi \approx \gamma_i \approx y_j = f(x_1, x_2, \ldots, x_m) + \varepsilon \quad 1 \leq m \leq M, \tag{1.2.3.3}$$

where the real world event result, ψ, approximated by the i^{th} iteration of the artificially simulated output, γ, that itself is now approximated by the meta-model response, y, a function f, of the M known number of factors represented as x_m, $1 \leq m \leq M$. In some cases the simulation input factor is identical to a regression meta-model variate. In other cases, the meta-model variate will be some sort of transformation on one or more of the simulation inputs. Kleijnen [61] provides an example of this case that describes a logarithmic ($x_1 = \log z_1$) or a non-linear ratio transformation on multiple simulation factors ($x_1 = z_1/z_2$). The deviation of the meta-model response y from the i^{th} simulated response γ_i is represented by $\gamma_i - y = \varepsilon$.

The content that follows, illustrate regression based and artificial neural net based meta-model methods that will explicitly construct the univariate and multivariate functional formulations of the abstract visualization in equation (1.2.3.3).

1.2.3.2 General Linear Regression Meta-model.

The generalized linear model is the most common meta-model method. Kleijnen has provided an extensive library of publications to include [59, 64, 60, 63], that provide theoretical and applied contributions to this branch of regression based meta-models. There are two levels of sophistication in regards to regression models in the form of single variate or multivariate linear regression models. It is intuitive here that single variate linear regression deals with situations where there is only one simulation response, while multivariate linear regression models deal with scenarios

where multiple simulation responses are being analyzed. The explicit construction of these methods are explained in greater details in the two sections that follow. The format found in Bauer [14] is followed.

1.2.3.3 Multiple Linear Regression Meta-model.

The simple linear regression model can be examined as

$$E[y|x] = b_0 + b_1 x \tag{1.2.3.4}$$

where the expected dependent variate y given the independent linear variate x is best approximated by the b_i coefficients. This formulation expands to the multiple linear regression model where there are multiple linear independent variates as is seen in Bauer's work [14].

Let X_i be the i^{th} observed, $1 \times (M+1)$ row vector (with a leading 1 for the intercept term) of deterministic input variates given as,

$$X_i = \begin{bmatrix} 1 & x_{i1} & x_{i2} & \cdots & x_{iM} \end{bmatrix}. \tag{1.2.3.5}$$

β be a $(M+1) \times 1$ column vector of estimating coefficients given as,

$$\beta = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_M \end{bmatrix}. \tag{1.2.3.6}$$

Now the i^{th} observed simulation response y_i is given as,

$$y_i = X_i\beta + e_i, \quad 1 \leq i \leq N, \tag{1.2.3.7}$$

where ε_i is the i^{th} observed residual. This assumes

$$y_i = E[y_i|X_i] + e_i, \tag{1.2.3.8}$$

where $e_i \sim N(0, \sigma^2)$, $E[y_i] = X_i\beta$, and $\text{var}[y_i] = \sigma^2$. Section 1.2.3.5, proves the minimizing β^* solution, via least squares, is given as

$$\beta^* = (X_i'X_i)^{-1}X_i'y_i, \tag{1.2.3.9}$$

in the generalized multivariate context.

1.2.3.4 Multivariate Linear Regression Model.

This section covers the case when the previous univariate simulation response formulation from equation (1.2.3.4) is extended to having multivariate simulation responses, now given as

$$Y_i = X_i\beta + \varepsilon_i, \quad 1 \leq i \leq K, \tag{1.2.3.10}$$

where X_i is the i^{th} observed $1 \times (M+1)$ row vector of deterministic input variates (with a leading 1 for the intercept term) given in equation (1.2.3.5), Y_i is the i^{th} observed $1 \times P$ row vector of simulation responses given as

$$Y_i = \begin{bmatrix} y_{i1} & y_{i2} & \cdots & y_{iP} \end{bmatrix}, \tag{1.2.3.11}$$

β is a $(M+1) \times P$ matrix of estimating coefficients given as

$$\beta = \begin{bmatrix} b_{01} & b_{02} & \cdots & b_{0P} \\ b_{11} & b_{12} & \cdots & b_{1P} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M1} & b_{M2} & \cdots & b_{MP} \end{bmatrix}, \tag{1.2.3.12}$$

and ε_i is the i^{th} observed $1 \times P$ row vector of residuals, given as

$$\varepsilon_i = \begin{bmatrix} e_{i1} & e_{i2} & \cdots & e_{iP} \end{bmatrix}. \tag{1.2.3.13}$$

Similarly to the single response situation, it is assumed $\varepsilon_i \sim N(0, \Sigma_i)$, $E[Y_i] = X_i\beta$, and $\text{var}[Y] = \Sigma_i$ when it is also assumed that X_i is deterministic.

The formulation from equation (1.2.3.10) above can be generalized across the replications and is given as

$$Y = X\beta + \varepsilon \tag{1.2.3.14}$$

where Y is a $N \times P$ matrix of simulation responses across N replications that is a generalized version of equation (1.2.3.11) given as

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1P} \\ y_{21} & y_{22} & \cdots & y_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{NP} \end{bmatrix}, \tag{1.2.3.15}$$

X is a $N \times (M+1)$ matrix (with a leading column of ones for the intercept terms) of input variates given as

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1M} \\ 1 & x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{NM} \end{bmatrix}, \tag{1.2.3.16}$$

β is $(M+1) \times P$ matrix of estimating coefficients given in equation (1.2.3.12), and ε is a $N \times P$ matrix of residuals given as

$$\varepsilon = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1P} \\ e_{21} & e_{22} & \cdots & e_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N1} & e_{N2} & \cdots & e_{NP} \end{bmatrix}. \tag{1.2.3.17}$$

The least squares solution of the minimized sums of squared error in this generalized case is resolved similarly as previously seen from equation (1.2.3.9) given as

$$\beta^* = (X'X)^{-1}X'Y, \tag{1.2.3.18}$$

Bauer [14] states the unbiased estimator for Σ^* is given as

$$\Sigma^* = \frac{(Y - X\beta^*)'(Y_X\beta^*)}{N - Q - 1} \tag{1.2.3.19}$$

1.2.3.5 Least Squares Optimality.

The method of least squares is used to find the optimal β^* that minimizes the sum of squared residuals. Equation (1.2.3.9) can be written as

$$\varepsilon = Y - X\beta. \tag{1.2.3.20}$$

Then sum of squared error (SSE) is given as

$$\text{SSE} = Y'Y - 2\beta'X'Y + \beta'X'X\beta. \tag{1.2.3.21}$$

(See equation (B.3.0.65) of Appendix B.3.1). The first derivative of equation (B.3.0.69) is given as

$$\frac{\partial SSE}{\partial \beta} = -2X'Y + 2X'X\beta. \tag{1.2.3.22}$$

Evaluating equation (1.2.3.22) at zero will be the minimizing solution given as

$$\beta = (X'X)^{-1}X'Y \tag{1.2.3.23}$$

(See equation (B.3.0.70) of Appendix B.B.3) for derivation. This is assuming that the inverse of $(X_i'X_i)^{-1}$ exists. The second derivative of equation (B.3.0.69) is given as

$$\frac{\partial^2 SSE}{\partial \beta \partial \beta'} = 2X'X, \tag{1.2.3.24}$$

a positive definite matrix which is a requirement for the minimizing solution to be β^* given as

$$\beta^* = (X'X)^{-1}X'Y. \tag{1.2.3.25}$$

In practice, the variance σ^* is unknown and therefore an unbiased estimator must be calculated which Bauer provides [14] and is given in equation (1.2.3.9).

1.2.3.6 Regression Meta-model with Control Variates.

The general linear regression meta-model that is examined in Section (1.2.3.2) are now augmented with controls variates. Two scenarios are considered here, univariate

and multivariate response meta-models. The last section will evaluate the unbiased estimators that is necessary for consideration when stochastic control variates are introduced to the regression model.

1.2.3.7 Multiple Control Variates for Multiple Population, Single Response Simulation.

Nozari et al. [92] extended the pure control variate theory to a multiple population design which falls under the context of meta-models. Nozari represented this design by a multiple linear regression model, where the standard input variates are considered to be deterministic with the addition of the stochastic control variates as they are sampled across the design points of an experimental design.

The basic formulation for this section extends equation (1.2.3.7) to be given as

$$y_i = X_i\beta + C_i\gamma + e_i, \tag{1.2.3.26}$$

where y_i is the i^{th} observed simulation response, X_i is the i^{th} observed $1 \times (M+1)$ row vector of deterministic input variates (with a leading 1 for the intercept term) given in equation (1.2.3.5), β is a $(M+1) \times 1$ column vector of estimating coefficients given in equation (1.2.3.6), C_i is the i^{th} observed, $1 \times Q$ row vector of stochastic control variates given as

$$C_i = \begin{bmatrix} c_{i1} & c_{i2} & \cdots & c_{iQ} \end{bmatrix}, \tag{1.2.3.27}$$

γ is a $Q \times 1$ column vector of estimating control coefficients given as

$$\gamma = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_Q \end{bmatrix}, \tag{1.2.3.28}$$

and e_i is the scalar valued i^{th} observed residual. In the multivariate response application (to be reviewed later), e_i takes on the form of ε to indicate a factor of residual values.

Since C_i is a vector of random values with known means that have had the known mean subtracted off, implies the $E[C_i] = 0$. We can assume then according to Bauer [14] that

$$\begin{pmatrix} Y_i \\ C_i \end{pmatrix} \sim N_{Q+1} \left(\begin{pmatrix} \mu_i \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2 & \Sigma_{YC} \\ \Sigma_{CY} & \Sigma_{CC} \end{pmatrix} \right) \tag{1.2.3.29}$$

where $E(Y_i) = \mu_i$ and Σ_{YC}, Σ_{CY} and Σ_{CC} are covariance matrices of the response and the controls and the controls with themselves, respectively.

1.2.3.8 Multiple Control Variates for Multiple Population, Multiple Response Simulation.

Porta Nova et al. [101] generalized the work by Nozari et al. [92] in Section 1.2.3.7 to simulations with multivariate simulation responses, amongst other findings.

Let Y be a $K \times P$ matrix of K vertically stacked replications of a $1 \times P$ vector, $Y_i = [Y_{i1}, Y_{i2}, \ldots, Y_{iP}]$, of simulation responses, and C be a $K \times Q$ matrix of K vertically stacked replications of a $1 \times Q$ vector, $C_i = [C_{i1}, C_{i2}, \ldots, C_{iQ}]$, of stochastic

control variates. We assume a joint normality between the simulation responses, Y, and the CVs, C, expressed in the matrix normal distribution as

$$\begin{bmatrix} Y \\ C \end{bmatrix} \sim \mathcal{MN}_{K,P+Q} \left(\begin{bmatrix} X \cdot \beta \\ 0_{K \times Q} \end{bmatrix}, \quad I_K, \quad \begin{bmatrix} \Sigma_Y & \Sigma_{YC} \\ \Sigma_{CY} & \Sigma_C \end{bmatrix} \right), \tag{1.2.3.30}$$

where X is a $K \times (M+1)$ matrix of K vertically stacked replications of a $1 \times (M+1)$ vector, $X_i = [1_i, X_{i1}, X_{i2}, \ldots, X_{iM}]$, of known deterministic input variables with a leading column of ones, β is a $(M+1) \times P$ matrix of unknown coefficients, and I_k is a $K \times K$ identity matrix indicating independence across the replications. Provided that $K > (M+1) + P + Q$, Porta Nova gives the meta-model as

$$Y = X \cdot \beta + C \cdot \Delta + R, \tag{1.2.3.31}$$

where Δ is a $Q \times P$ matrix of unknown coefficients, and R is a $K \times P$ matrix of K vertically stacked $1 \times P$ vectors, $R_i = [R_{i1}, R_{i2}, \ldots, R_{iP}]$, of residuals. See proposition 3 in the Appendix of [99] for explanation of the point estimators for β and Δ, given as

$$\widehat{\beta} = (X' \cdot X)^{-1} \cdot X' \cdot [I - C \cdot (C' \cdot P \cdot C)^{-1} \cdot C' \cdot P] \cdot Y, \tag{1.2.3.32}$$

and

$$\widehat{\Delta} = (C' \cdot P \cdot C)^{-1} \cdot C' \cdot P \cdot Y, \tag{1.2.3.33}$$

where

$$P = I - X \cdot (X' \cdot X)^{-1} \cdot X'. \tag{1.2.3.34}$$

Also see Section 3.7 on pg. 54 of Seber & Lee [112] for analysis on the more generalized context of introducing further explanatory variables to a general linear regression model. Porta Nova et al. [101] proves the $100(1-\alpha)\%$ confidence ellipsoid about vec$[\beta]$ is formed from

$$Pr\left[\frac{K-m-Q-mp+1}{mp(K-m-Q)} \leq F_{mp,K-m-Q-mp+1}(1-\alpha)\right] = 1-\alpha \tag{1.2.3.35}$$

Porta Nova [101] provides a loss factor of

$$\lambda_3 = \left[\frac{K-m-1}{K-m-Q-1}\right]^p, \tag{1.2.3.36}$$

in addition to the minimum variance ratio as

$$\eta_3' = \left[\prod_{i=1}^{v}(1-\rho_i^2)\right]^m. \tag{1.2.3.37}$$

1.2.3.9 Neural Network Meta-model.

Simulation modeling attempts to reproduce the exact nature of a real world system in an effort to capture the relationships and provide a safe environment for various kinds of environments that can extrapolate new solutions. Unfortunately, many times

the computational and monetary expenses to run these simulations can be exorbitant and cumbersome for quick decisive action at the decisional support layer. One common effort is to to find optimal settings that will allow the real world system to achieve some predetermined goal or objective. However, it would be necessary to rerun a simulation many, many times to find an acceptable solution. Furthermore, when quick decisions need to be made, simulations are oftentimes take too long to run and analyze to support leadership in such a fashion. To supplement these simulations, a practitioner can construct an artificial neural network to approximate the outputs of the simulation and quickly find optimal solutions from the non-linear function the neural network generates using common optimization techniques like evolutionary algorithms to quickly find approximations of the simulation. Another advantage that neural networks offer is the agility in handling different types of domains from continuous to noncontinuous valued functions. Overall, neural networks have proven to be a promising mechanism to employ in replacement of other classical techniques like polynomial regression models that can only model small regional sections of a functions surface. Padgett & Roppel [96] have stated, "Neural networks in all categories address the need for rapid computation, robustness, and adaptability. Neural models require fewer assumptions and less precise information about the systems modeled than do some more traditional techniques."

Kilmer [55] suggests starting with a baseline model that, "...involves using a back-propagation trained, multi-layer artificial neural network to learn the relationship between the simulation inputs and outputs." He continues with a proposed three phased approach to determine the relevant decision and response variables of interest, run the simulation to gather training samples and train the neural net, and finally validate the neural nets value by taking simulated output from the artificial neural net and running them back in the original computer simulation. However, this

marginalizes the finer details of constructing an artificial neural network topology that can approximate all the relationships of a computer simulation. Many considerations have to be made in the design process before confidence can be formed about the network's structure properly estimating the underlying simulation. Li et al. [71] provides three predetermined parameters of a baseline neural network model, that can play a significant part in the performance of the net, "among these parameters, one hidden layer is often recommended as multiple hidden layers may lead to an over parameterized ANN structure."

Some additional points of interest are by Kilmer [54] that took a two input inventory model illustrated by Law [70] and applied a neural network to estimate a mean total cost. Hurrion [46] built a multiple response neural network meta-model for a stochastic simulation that models coal depot factory, that outputs the mean and 99% confidence interval. Badiru et al. [7] used a neural network that input the initial investment and interest rate information as random variables to predict the future cash flow for cost estimation. Pierreval et al. [98, 81] provided research on neural networks as a meta-modeling tool for discrete-event and continuous simulation models, and more specifically with dispatching problems.

A looming open research item is the application of artificial neural networks in stochastic simulations. There has been some interesting work by Kilmer [55] in estimating underlying statistical parameters like mean and variance using individual neural networks. And beyond the standard formulation of the artificial neural network that is described above, there exist far more exotic and elaborate designed neural networks (e.g., convolutional, deep learning, recurrent, etc.) that can be invoked in applied settings. The reliance on meta-modeling has steadily increased due to declining budgets and the potential cost savings in using meta-modeling as a surrogate for more robust and expensive simulation runs. Artificial neural networks have proven

to be a bona fide solution for high fidelity approximations of large scale computer simulations.

1.2.4 Jackson Network Analysis for Open Networks with M/M/s/∞ queues.

An inter-connected network of queues is described by having an ith node represented by a service station of s_i servers with mean service time μ_i, and infinite capacity queue. This network is classified as a Jackson network if the arrival of entities from outside the network can be treated as a poisson process with Λ inter-arrival mean, if all service stations can be treated as an independent and exponentially distributed stochastic process, and if after an entity traverses a random path through the network of nodes it eventually departs the network. This special type of connected network of queues that is characterized by the Jackson Network allows for feedback loops to exist allowing for repeating stations consecutively or moving to a preceding service station that was previously visited in its path.

The total arrival rate of internal and external entities to the ith queue of a Jackson network is defined by $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]$ as a $1 \times N$ row vector of queue network traffic equations, and given as

$$\lambda = \Lambda + \lambda \cdot P, \tag{1.2.4.1}$$

where $\Lambda = [\Lambda_1, \Lambda_2, \ldots, \Lambda_N]$ is a $1 \times N$ row vector of external entity arrival rates to the ith queue, and P is a $N \times N$ matrix of probabilities of routing an internal entity from the ith queue to a subsequent jth queue in the network, given as

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{N,N} \\ p_{2,1} & p_{2,2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ p_{N,1} & \cdots & \cdots & p_{N,N} \end{bmatrix}. \tag{1.2.4.2}$$

If I is a $N \times N$ identity matrix and assuming $(I - P)$ is invertible, a unique solution can be resolved for λ, by

$$\lambda = \Lambda(I - P)^{-1}. \tag{1.2.4.3}$$

The equilibrium of the queue network is defined by $\rho = [\rho_1, \rho_2, \ldots, \rho_N]$ as a $1 \times N$ row vector of inequalities that determine if the ith queue node is stable, that is, if

$$\rho = \lambda \cdot (S \circ \mu)^{-1} < 1, \tag{1.2.4.4}$$

where $S = [s_1, s_2, \ldots, s_N]$ is a $1 \times N$ row vector of the number of server resources for each of the N queues in the network, $\mu = [\mu_1, \mu_2, \ldots, \mu_N]$ is the mean service time for each of the N queues in the network, and λ is provided by equation (1.2.4.3).

Providing results from Gross et al. [39], pp. 68-72 & 190-191, the long run mean system sojourn time for entities entering network is given as

$$W = (1^t \cdot L) \cdot (1^t \cdot \Lambda)^{-1}, \tag{1.2.4.5}$$

where 1^t is a $N \times 1$ column vector of ones, Λ is given in equation (1.2.4.1), and L is a $1 \times N$ row vector of the average number of entities at each of the N nodes in the network.

For comprehensive analysis of Jackson networks see, Gross et al. [39], Kulkarni [65], and Bose [19].

1.2.5 Interior Point method for Non-linear Convex Optimization.

Bazaraa [17] gives the general form for a non-linear optimization problem as,

$$\begin{aligned} Minimize \quad & f(x) && (1.2.5.1)\\ \text{subject to} \quad & g(x) \leq 0 && (1.2.5.2)\\ & h(x) = 0 && (1.2.5.3) \end{aligned}$$

Convex optimization is a special class of non-linear optimization problems, when both of the objective and constraint functions in the optimization problem are convex. This means it satisfies the following inequality provided by Bazaraa [17].

"Let $f : S \to R$, S is a nonempty convex set in R_n. The function f is said to be *convex* on S if

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2) \qquad (1.2.5.4)$$

for each $x_1, x_2 \in S$ and for each $\lambda \in (0, 1)$."

Conventionally, this means a function is said to be convex if and only if it contains the line segment between any two points on the graph of the function. This is generalized in Figure 11 below, of a two-dimensional feasible region where the line segment between points x_1 and x_2 are entirely feasible.

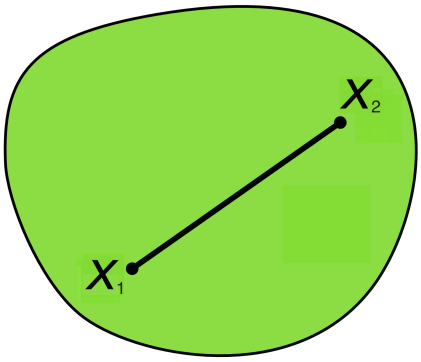


Figure 11. Two Dimensional Convexity Example

Intuitively, non-convexity would be any optimization problem when the objective function or constraint function(s) violate the definition of equation (1.2.5.4) above. This is generalized in Figure 12 below, of a two-dimensional feasible region where the entire line segment between points x_1 and x_2 is *not* contained (when red segment of line lies outside the boundary of the green colored area) in the graph of the function.

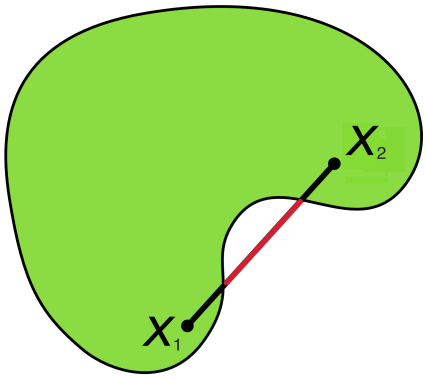


Figure 12. Two Dimensional Non-Convexity Example

The interior point method is one of many types of routines to optimize this class of optimization problems. This technique traverses through a central path of a feasible region, by augmenting the objective function with a barrier function. This is

fundamentally different from the simplex method, which traverses the extreme points of a feasible region, from one vertex to an adjacent vertex. This is also different then the active set method that traverses through the edges of the feasible region. The interior point method converts a constrained non-linear objective function into an unconstrained problem by transforming the constraints, which uses barrier functions for inequality constraints, and Lagrange multipliers for equality constraints.

Referring back to the general formulation of a non-linear optimization problem, Byrd et al. [21] transforms equations (1.2.5.1)-(1.2.5.3) by introducing an $m \times 1$ vector of slack variables, giving

$$\begin{aligned} Minimize \quad & f(x) && (1.2.5.5) \\ \text{subject to} \quad & g(x) + s = 0 && (1.2.5.6) \\ & h(x) = 0 && (1.2.5.7) \\ & s \geq 0. && (1.2.5.8) \end{aligned}$$

This transformation, replaces the inequality constraints with the equality bounds associated with the slack variables, $s \geq 0$. Further, Byrd [21] transforms this slack variable formulation of the original general form, a second time to, giving

$$\begin{aligned} Minimize \quad & f_\mu(x) - \mu \cdot B(x) && (1.2.5.9) \\ \text{subject to} \quad & s \geq 0. && (1.2.5.10) \end{aligned}$$

where the "barrier function" (also called a "merit function" in Matlab [75]) is,

$$B(x) = \sum_{i=1}^{m} \ln s_i, \tag{1.2.5.11}$$

μ is a barrier parameter, and s is assumed to be positive as defined by Byrd [21]. Now, as μ approaches zero, the minimum of f_μ should decrease to the minimum of f. There are many approaches for minimizing f_μ, Byrd et al. [22, 21] has combined ideas from interior point and trust region methods, or with sequential quadratic programming, respectively, to allow the direct use of second order derivatives. Waltz et al. [123] also combines interior point methods with line search methods that switches between newton's method and a conjugate gradient step "that guarantees progress toward stationarity...", using trust region.

The approach used for interior point optimization in this research (from built-in interior point optimization routine, *fmincon*, in Matlab [75]) is similar to Waltz'. It is a two step-routine, with the first attempt being a direct step through a linear approximation (Newton's method) that could resolve the Karush-Kuh-Tucker (KKT) conditions of optimality. If these conditions fail, when the projected Hessian is not positive definite, then the method falls back to a conjugate gradient step, which Waltz et al. [123] states "guarantees progress toward stationarity...". If the conjugate gradient step is invoked, the algorithm evaluates Lagrange multipliers by solving the KKT conditions for the quadratic approximation in a trust region.

See Bazaraa [17] text for complete coverage of the interior point method (pp. 501-502), or for the many other types of convex and non-convex non-linear optimization methods.

1.2.6 TV Inspection and Adjustment Stations on a Production Line Simulation for Research Objectives.

A simulation that was provided by Pritsker et al. [104] (pp. 108-114) and illustrated in Figure 13 below, is used exclusively for the research of this dissertation. This system emulates the arrival of assembled TVs into the system at the inspection station. If a TV fails inspection, it is routed to the adjusting station for adjustment and then returned through a feedback loop to be re-inspected at the inspection station. This continues until the TV passes inspection and exits the system.

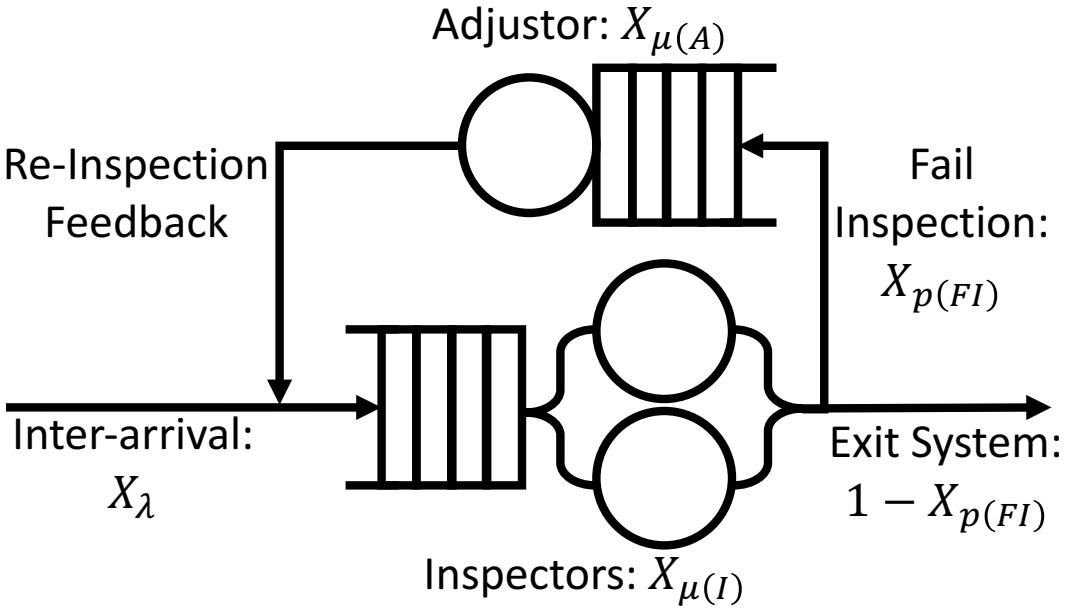


Figure 13. Diagram of TV Inspection & Adjusting System [16]

This simulation meets the criteria to be characterized as an open Jackson network that is defined by Subsection 1.2.4. The inter-arrival process of the TVs is modeled by a Poisson process. The two service stations are modeled as M/M/s/∞ queues, where $s = 2$ at the inspection station and $s = 1$ at the adjusting station.

1.3 Research Overview

This dissertation is organized as follows. Chapter II extends the statistical framework for variance reduction for the direct estimation of single response, single population simulations, by including second order interactions of control variates. It is shown this extension retains the properties associated with unbiased estimators. Chapter III extends the statistical framework for least squares regression meta-models for single response, single population simulations, by including second order interactions between control variates, and second order interactions between control variates and the designed input variables. It is shown this extension also retains the properties associated with unbiased estimators. Chapter IV, extends the conventional formulation of the radial basis neural network meta-model for simulations, to now include control variates. Chapters II and Chapters III also demonstrate the heightened insight into the system dynamics that are governing the simulation response of interest. Finally, Chapters III through IV demonstrate through similar extensive case studies, that the new extensions made in each chapter, significantly outperform their respective standard counterparts that is common in literature. Chapter V, summarizes the original research contributions and prospects for future research.

II. Unveiling System Dynamics for Single Population Simulations Using Linear Control Variates with Interactions

2.1 Introduction

The origin for control variates (CVs), also referred to as regression sampling or concomitant variables, can be traced back to their use in Monte Carlo theory. Wilson [129] comprehensively surveyed and developed a two-category taxonomy for variance reduction techniques (VRTs), for which the CV method joins common random numbers and antithetic variates as components of the correlation based category. Since the introduction of CVs to simulation in the early 1980s, a well regarded body of knowledge for CVs has developed over the past 35+ years that includes many practical applications to a wide variety of fields of study that demonstrate its robust utility for simulation. In most cases and when compared to other approaches, the computational expense of internalizing CVs is negligible while the latent potential to reduce variance is superior. Fundamental CV theory for simulation rests primarily upon four important pieces of literature. These publications contextualize the aptitude of CVs to reduce variance for direct estimation of univariate and multivariate response simulations (see Lavenberg & Welch [69] and Rubenstein & Marcus [109], respectively), and for linear metamodels of univariate and multivariate simulations (see Nozari, Arnold, & Pegden [92] and Porta Nova & Wilson [101], respectively).

Two subsequent articles examine new types of CVs that induce properties favorable to the context of this research. Wilson & Pritsker [132] provide analysis for what they call "work" variables that are characterized as asymptotically standardized CVs. Bauer & Wilson [16] introduced "routing" CVs that are evaluated from the stochastic branching processes commonly found in simulations that are similarly characterized

as asymptotically standardized CVs. These two types of CVs treat a large proportion of the stochastic processes found in most simulations.

Upon thorough inspection, it is clear that the collective research on CVs has focused almost exclusively on constructing increasingly sophisticated prescriptions for the use of CVs that yield improvements to statistical estimates for simulation. Nelson [87] acknowledged a need for the community at large to stimulate CV research in new directions, reasoning for alternatives that go beyond enhancing variance reduction "because much of the linear CV theory has been treated". This chapter finds common ground with Nelson's finding which pivots away from this aforementioned predisposition and towards an entirely new direction that will illuminate the underlying system dynamics for simulation.

First introduced in 1961 by Forrester [34], system dynamics is described by Coyle [27] as a discipline that seeks to construct interdisciplinary frameworks that can identify and articulate what governs a complex system of interest, or for the purposes of this research, a stochastic simulation. System dynamic constructs are woven from an extensive literature that document the assumptions and conditions that are favorable and necessary for the discovery of the components that govern a simulation, see Ogata [93], Karnopp et al. [53], & Palm [97]. Starting in 1983, the System Dynamics Society was formed and has since documented much of this literature, see surveys by Forrester [33] on broader system dynamic thinking and Kleijnen [57] for more specific context under sensitivity analysis and designed experiments (DOE). Kleijnen [61] further illustrates the use of DOE, regression, and VRTs as a foundation for employing sensitivity analysis to identify important factors in complex simulations. The manifestation of the system dynamics in Kleijnen's case, is observed from the sensitivity of the simulation response(s) as it is replicated across a DOE prescribed sequence of settings. This chapter, however, proposes observing the sensitivity of the simula-

tion response(s) from the perturbations of the linear and second order interactions of "work" and "routing" CVs. This is an important distinction as this requires only a single replicated setting to extract knowledge about the underlying system dynamics of the simulation.

This new direction for CVs will, under certain conditions, preserve the unbiased nature of the CV estimator and maintain its utility to significantly reduce variance. These ideas are demonstrated with the empirical results from a two-part case study which exhibit a more lucid picture of the underlying system dynamics for a simulation when second order interactions of multiple types of CVs are internalized as additional regressors. The motivation for this research is based on improving a decision maker's understanding of a complex simulation that is equivalently, if not greater, in value than improving its statistical estimates.

The organization for the remaining material of this chapter is as follows. Section 2 will provide the extended statistical framework that is woven from several of the articles mentioned above. Section 3 examines the methodology that was undertaken for the case study. Section 4 presents findings from the case study. Section 5 will summarize and conclude this chapter.

2.2 Control Variate Model with Interactions

Let $W = [W_1, W_2, \ldots, W_Q]'$ be a $Q \times 1$ column vector of standardized "work" CVs and $R = [R_1, R_2, \ldots, R_P]'$ be a $P \times 1$ column vector of standardized "routing" CVs, then $Z = [W_1, W_2, \ldots, W_Q, R_1, R_2, \ldots, R_P]'$ can be defined as a $(Q + P) \times 1$ column vector of vertically concatenated "work" and "routing" CVs. Now let $\delta = [\delta_1, \delta_2, \ldots, \delta_{(Q+P)}]$ be a $1 \times (Q + P)$ row vector of linear estimating coefficients and Δ be a $(Q + P) \times (Q + P)$ matrix of second order interaction and quadratic estimating coefficients given as

$$\Delta = \begin{bmatrix} \Delta_{1,1} & \Delta_{1,2} & \cdots & \Delta_{1,(Q+P)} \\ \Delta_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Delta_{(Q+P),1} & \cdots & \cdots & \Delta_{(Q+P),(Q+P)} \end{bmatrix}. \tag{2.2.0.1}$$

The second order interaction and quadratic terms are appended to equation (1.2.1.1), giving us

$$y(\delta, \Delta) = y - \delta \cdot Z - Z' \cdot \Delta \cdot Z. \tag{2.2.0.2}$$

However, asymptotically standardized "work" and "routing" CVs are used, whcih are all independent from each other as elements of Z, this implies equation (2.2.0.2) is biased since $E\{Z' \cdot \Delta \cdot Z\} = \text{tr}\{\Delta\}$. See pg. 46 of Anderson [5] for moments and cumulants of jointly normal variants. Now, $\tilde{\Delta} = \Delta - \text{diag}\{\Delta\}$ is defined, that is to mean, place zeros as the estimating coefficient for all second order quadratic effects associated with $\text{diag}\{Z'Z\}$ then $\tilde{\Delta}$ can be substituted for Δ into equation (2.2.0.2), then the model

$$y(\delta, \tilde{\Delta}) = y - \delta \cdot Z - Z' \cdot \tilde{\Delta} \cdot Z, \tag{2.2.0.3}$$

is an unbiased estimate for y.

2.3 Case Study

This section will summarize a comprehensive two part case study that was conducted to evaluate the utility of internalizing the interactions across the standardized "work" & "routing" CVs. The first part of this study, in 2.3.1, demonstrates that unbiased second order CV terms can be included without significantly affecting the capacity to reduce variance under favorable conditions. Thereafter, the second intent in 2.3.2, demonstrates that these additional terms can unveil a better understanding of the system dynamics of the simulation, more so than could be extracted from a first-order CV model.

2.3.1 TV Inspection and Adjustment Production System.

The simulation used in this research, meets the criteria to be characterized as an open Jackson network that is defined by Subsection 1.2.6. The inter-arrival process of the TVs is modeled by a Poisson process. The two service stations are modeled as M/M/s/∞ queues, where $s = 2$ at the inspection station and $s = 1$ at the adjusting station. The long run mean sojourn time for the TVs in the system is the primary response of interest.

The input settings that were held constant across all configurations were the:

- X_{λ}: TV inter-arrival time with mean of $1/\lambda = 10.0$ time units, the
- $X_{\mu(I)}$: inspection service time with mean of $1/\mu(I) = 6.0$ time units, and the
- $X_{\mu(A)}$: adjustor service time with mean of $1/\mu(A) = 20.0$ time units.

The input variable that was changed across the configurations was the:

- $X_{p(FI)}$: branching process for failing inspection with mean of $p(FI) = \{.05, .10, \ldots .15, .20, .25, .30, .31, .32, .33, .333\}$.

2.3.2 Part I: Variance Reduction with Non-terminating Simulation.

The true long run mean sojourn time in system for each configuration of this simulation was evaluated using the Jackson network equations (1.2.4.1) - (1.2.4.5) presented in Subsection 1.2.4. Table 1 summarizes the results of these computations, $W_{Jackson}$, as well as the estimates, $\widehat{W}_{G\&B}$, that are computed using 1,000 replications from each of the inspection failure rate, $X_{p(FI)}$, configurations. As part of an effort to minimize transient bias, sufficient total run length and steady state truncation points were established for each of the configurations that resulted in the collection of statistics that suggested the transient bias had been eliminated. Table 1 also contrasts the results from the configurations ($X_{p(FI)} \leq 0.25$) that overlap with the configurations that were evaluated and reported by Bauer & Wilson [16], $\widehat{W}_{B\&W}$, that include their computations from Queueing Network Analyzer (QNA) [126], W_{QNA}. It is noted, Bauer & Wilson [16] evaluated configurations for $X_{p(FI)} = \{0.05, 0.10, 0.15, 0.20, 0.25\}$, excluding configurations $X_{p(FI)} > 0.25$.

Table 2. Average TV sojourn time in system

	Stop	Steady	Stability		Gibb & Bauer		Bauer & Wilson	
$X_{p(FI)}$	Sim.	State	$\rho_{\mu(I)}$	$\rho_{\mu(A)}$	$W_{Jackson}$	$\widehat{W}_{G\&B}$	W_{QNA}	$\widehat{W}_{B\&W}$
0.050	40k	20k	0.316	0.105	8.192	8.189	8.192	8.185
0.100	40k	20k	0.333	0.222	10.357	10.350	10.357	10.341
0.150	40k	20k	0.353	0.353	13.518	13.550	13.518	13.480
0.200	40k	20k	0.375	0.500	18.727	18.717	18.727	18.627
0.250	150k	75k	0.400	0.667	29.524	29.516	29.524	29.430
0.300	300k	150k	0.429	0.857	70.500	70.307		
0.310	300k	150k	0.435	0.899	99.294	99.014		
0.320	400k	200k	0.441	0.941	170.956	170.819		
0.330	1.5M	750k	0.448	0.985	671.201	671.836		
0.333	700M	350M	0.450	0.999	6,671.300	6,679.198		

Two additional characteristics that were varied in this study that are necessary for evaluating the $100(1-\alpha/2)\%$ confidence intervals (with $\alpha = 0.9$) for the long run mean TV sojourn time in system, was the:

- sample set size = $\{10, 20, 100, 250\}$, and the
- selected subset of CVs (see Table 3) = $\{(1), w, R, wR, wRi, WR, WRi\}$

For w, R, wR, & wRi, the "work" CV for the inter-arrival stochastic process were *excluded*, W_{λ}, as part of an effort to contrast the results with Bauer & Wilson [16]. For WR and WRi, this "work" CV for the inter-arrival stochastic process is *included* as an extension to the prior research.

Table 3. Subsets of CVs ($X \equiv$ term included)

	(1)	w	R	wR	wRi	WR	WRi
1: W_λ						X	X
2: $W_{\mu(I)}$		X		X	X	X	X
3: $W_{\mu(A)}$		X		X	X	X	X
4: $R_{p(FI)}$			X	X	X	X	X
1×2							X
1×3							X
1×4							X
2×3					X		X
2×4					X		X
3×4					X		X

To evaluate the variance reduction efficiency for the different sets of CVs, meta-experiment tactic was deployed and also detailed by Bauer & Wilson [16]. A total of 240 meta-experiments were conducted (4 sample sizes $\times$ 10 different probabilities of failing inspection $\times$ 6 subsets of CV, including no CVs). For each of these meta-experiments, two measures were considered: 1) confidence interval coverage, and 2) variance reduction. Equations (1.2.1.40) - (1.2.1.43) of Subsection 1.2.1.11 and equations (1.2.1.36) - (1.2.1.39) of Subsection 1.2.1.10 were used to construct the confidence intervals for each meta-experiment.

In Tables 4, 5, and 6, the results of the first part of this study are presented (also see Tables 21 and 22 for additional information). Table 4 provides the realized coverages across the number of confidence intervals (1,000 replications / selected sample size) for each of the 240 meta-experiments. Note that there do not appear to be significant

effects due to the three components of the study (i.e., branching probability, sample size, and CV subset) that are reconfigured.

Table 5 gives the average percent of variance reduction for the confidence interval half-length constructed with one of the six subsets of CVs, relative to the confidence interval evaluated without using CVs (using the same subsets of data). Table 6 gives the average percent of variance reduction for just the standard error component of the confidence intervals in Table 5.

Excluding the configurations where $X_{p(FI)} = 0.050$, its noted that it appears the variance reduction for the confidence intervals (in Table 5) and its standard error component (in Table 6), using only "routing" CVs, R, consistently outperformed the intervals that use only "work" CVs, w. Further, for the most part, the joint use of both "work" and "routing" CVs, wR, offers even better variance reduction than using either "work" or "routing" CVs exclusively. However, when the sample size is 10 and $X_{p(FI)} > 0.250$ the joint use of both types of CVs is outperformed by the intervals using "routing" CVs only, R.

Finally, it is interesting to note that as the sample size increases, the variance reduction for the intervals constructed with interaction terms, wRi and WRi, improve rapidly to be near the performance of the best CV models that excluded CV interaction terms, wR and WR. Moreover, as the probability of failing inspection is near or at its greatest value, these CV interaction models consistently provided the best variance reduction performance (+1 to +2%) of all six subsets of CVs (marked by an asterisk in Tables 5 and 6).

Table 4. Realized coverage [%]

$X_{p(FI)}$	Sample Size	(1) $B^{\dagger}$	(1) $G^{\dagger\dagger}$	w $B^{\dagger}$	w $G^{\dagger\dagger}$	R $G^{\dagger\dagger}$	wR $B^{\dagger}$	wR $G^{\dagger\dagger}$	wRi $G^{\dagger\dagger}$	WR $G^{\dagger\dagger}$	WRi $G^{\dagger\dagger}$
0.050	10	90	94	88	93	88	88	89	87	94	
0.100	10	90	90	86	90	87	86	89	96	89	
0.150	10	89	80	89	89	80	85	85	84	87	
0.200	10	88	90	86	88	91	81	85	91	87	
0.250	10	88	89	86	89	95	83	93	95	91	
0.300	10		89		91	92		92	92	93	
0.310	10		84		92	80		86	84	89	
0.320	10		85		79	83		76	82	78	
0.330	10		85		85	85		83	89	84	
0.333	10		85		85	86		86	90	85	
0.050	20	90	86	89	86	88	90	94	88	86	86
0.100	20	91	94	87	88	90	89	86	94	86	96
0.150	20	91	88	87	94	76	84	86	84	84	82
0.200	20	87	96	86	98	94	81	92	90	90	90
0.250	20	89	88	86	88	94	86	92	94	90	84
0.300	20		92		94	88		92	90	90	82
0.310	20		80		90	78		88	84	88	86
0.320	20		90		90	86		84	84	94	90
0.330	20		92		86	90		88	86	84	84
0.333	20		82		84	84		86	90	92	94

† Bauer & Wilson [16] †† Gibb & Bauer

Table 4 (cont.): Realized coverage [%]

$X_{p(FI)}$	Sample Size	(1) $B^{\dagger}$	(1) $G^{\dagger\dagger}$	w $B^{\dagger}$	w $G^{\dagger\dagger}$	R $G^{\dagger\dagger}$	wR $B^{\dagger}$	wR $G^{\dagger\dagger}$	wRi $G^{\dagger\dagger}$	WR $G^{\dagger\dagger}$	WRi $G^{\dagger\dagger}$
0.050	100		100		90	90		80	90	80	90
0.100	100		100		100	90		90	90	90	90
0.150	100		80		70	90		80	80	80	70
0.200	100		90		90	100		100	100	100	100
0.250	100		90		90	90		90	90	90	90
0.300	100		100		100	90		100	100	100	100
0.310	100		70		80	80		90	90	90	90
0.320	100		80		80	80		80	70	90	90
0.330	100		100		100	90		90	80	100	90
0.333	100		90		90	100		90	100	90	90
0.050	250		100		100	75		100	100	100	100
0.100	250		100		100	100		100	100	100	100
0.150	250		100		100	75		75	75	75	75
0.200	250		100		75	100		100	100	100	100
0.250	250		75		75	100		75	75	75	75
0.300	250		100		100	100		100	100	100	100
0.310	250		100		100	100		100	100	100	100
0.320	250		100		75	100		100	100	100	100
0.330	250		75		75	75		75	75	75	75
0.333	250		75		100	75		75	75	75	75

† Bauer & Wilson [16] †† Gibb & Bauer

Table 5. Average variance reduction relative to confidence interval w/o CVs [%]

$X_{p(FI)}$	Sample Size	w $B^{\dagger}$	w $G^{\dagger\dagger}$	R $G^{\dagger\dagger}$	wR $B^{\dagger}$	wR $G^{\dagger\dagger}$	wRi $G^{\dagger\dagger}$	WR $G^{\dagger\dagger}$	WRi $G^{\dagger\dagger}$
0.050	10	24	15	11	52	52	10	46	
0.100	10	16	4	17	47	42	-23	35	
0.150	10	13	-3	20	42	31	-21	25	
0.200	10	10	0	16	35	27	-41	21	
0.250	10	9	-3	5	29	8	-88	-2	
0.300	10		-9	9		3	-79	-3	
0.310	10		-10	11		4	-116	-3	
0.320	10		-9	9		2	-83	-5	
0.330	10		-9	5		-4	-95	-6	
0.333	10		-6	5		0	-86	-3	
0.050	20	28	25	14	57	58	51	58	41
0.100	20	18	15	21	53	53	48	53	35
0.150	20	15	9	24	46	42	33	42	13
0.200	20	12	10	20	37	38	27	39	14
0.250	20	10	6	12	31	24	12	23	-8
0.300	20		2	13		17	4	19	-9
0.310	20		2	14		20	5	21	-15
0.320	20		3	12		18	7	18	-22
0.330	20		1	10		13	4	16	-12
0.333	20		3	10		15	6	19	-12

† Bauer & Wilson [16] †† Gibb & Bauer

Table 5 (cont.): Average variance reduction relative to confidence interval w/o CVs [%]

		Subsets of CVs							
	Sample	w	w	R	wR	wR	wRi	WR	WRi
$X_{p(FI)}$	Size	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$
0.050	100		29	18		61	61	63	61
0.100	100		21	24		57	56	58	57
0.150	100		15	24		47	46	48	47
0.200	100		14	23		43	42	45	42
0.250	100		11	15		31	30	33	31
0.300	100		9	14		25	25	28	26
0.310	100		8	16		27	26	30	28
0.320	100		7	14		23	24*	27	28*
0.330	100		5	11		17	18*	20	21*
0.333	100		8	11		21	23*	25	27*
0.050	250		30	18		62	62	63	63
0.100	250		22	24		57	57	59	58
0.150	250		15	25		47	47	49	48
0.200	250		15	23		44	44	46	45
0.250	250		12	15		32	31	34	33
0.300	250		10	15		26	27*	29	30*
0.310	250		9	17		28	29*	31	32*
0.320	250		8	14		23	24*	28	31*
0.330	250		6	10		18	19*	21	23*
0.333	250		9	11		21	23*	26	29*

† Bauer & Wilson [16] †† Gibb & Bauer

Table 6. Average variance reduction relative to standard error of the confidence interval w/o CVs [%]

		Subsets of CVs							
	Sample	w	w	R	wR	wR	wRi	WR	WRi
$X_{p(FI)}$	Size	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$
0.050	10		18	12		54	30	51	
0.100	10		7	18		45	4	41	
0.150	10		0	22		35	6	32	
0.200	10		3	17		31	-10	28	
0.250	10		0	7		14	-47	7	
0.300	10		-6	11		8	-40	6	
0.310	10		-6	12		10	-69	6	
0.320	10		-5	11		7	-42	4	
0.330	10		-6	7		2	-52	3	
0.333	10		-3	7		6	-45	7	
0.050	20		25	14		58	52	59	45
0.100	20		16	21		53	49	54	38
0.150	20		10	24		42	34	42	18
0.200	20		11	20		39	29	40	19
0.250	20		7	12		25	14	24	-2
0.300	20		3	13		17	6	20	-3
0.310	20		3	14		21	8	22	-8
0.320	20		3	13		19	9	19	-15
0.330	20		1	10		13	6	17	-5
0.333	20		4	10		16	8	20	-5

† Bauer & Wilson [16] †† Gibb & Bauer

Table 6 (cont.): Average variance reduction relative to standard error of the confidence interval w/o CVs [%]

		Subsets of CVs							
	Sample	w	w	R	wR	wR	wRi	WR	WRi
$X_{p(FI)}$	Size	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$
0.050	100		29	18		61	61	63	61
0.100	100		21	24		57	56	58	57
0.150	100		15	24		47	46	48	47
0.200	100		15	23		43	42	45	43
0.250	100		11	15		31	30	33	31
0.300	100		9	14		25	25	28	26
0.310	100		8	16		27	26	30	28
0.320	100		7	14		23	24*	28	29*
0.330	100		5	11		17	18*	20	21*
0.333	100		8	11		21	22*	25	28*
0.050	250		30	18		62	62	63	63
0.100	250		22	24		57	57	59	58
0.150	250		15	25		47	47	49	48
0.200	250		15	23		44	44	46	45
0.250	250		12	15		32	31	34	33
0.300	250		10	15		26	27*	29	30*
0.310	250		9	17		28	29*	31	32*
0.320	250		8	14		23	24*	28	31*
0.330	250		6	10		18	19*	21	23*
0.333	250		9	11		21	23*	26	29*

† Bauer & Wilson [16] †† Gibb & Bauer

Intuitively, it is anticipated that increasing the probability of failing inspection, $X_{p(FI)}$, would push the stability of the adjusting station queue closer to a complete queue meltdown, $rho_{\mu(A)} >= 1$. When $X_{p(FI)} \geq 0.30$, the much stronger dependencies occurring between the stochastic processes in this small, but sufficiently complex simulation made it more difficult for the smaller first order CV models to sustain variance reduction performance. The disparate patterns of variance reduction that manifested in the "work" and the "routing" CVs suggests that these behaviors are responding to the changing system dynamics of the simulation.

Further, the superior performance of the CV interaction models, wRi and WRi, when the stability of the adjusting station queue was weak, support the notion that a second order CV model could provide a better model of the system variance, provided there is a sufficient number of replications. Otherwise, when there is not a sufficient number of replications, employing CV selection strategies is recommended, like Bauer & Wilson [15] who proposed a method for selecting CVs that are assumed jointly normal with the simulation response, by reducing the mean square volume of the confidence region for multi-response simulations. These revelations gave us a peek at the governing system dynamics of the simulation and portends the need to extrapolate the intuition into the second part of this study, which is presented in Subsection 2.3.3.

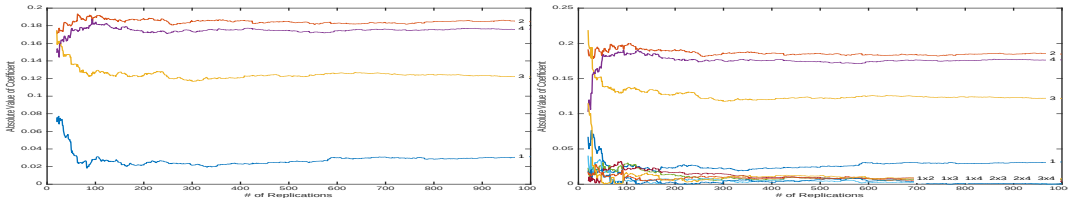
2.3.3 Part II: Unveiling System Dynamics with Non-terminating Simulation.

The objective for this second part of the study is to identify the important components of the simulation it was systematically re-configured to induce instability into the adjusting station queue. The two largest subsets of CVs, WR and WRi, was selected to be retained in this part of the study because their variance reduction performance was the best when provided sufficient replications. The first experiment in

this study looks at how the magnitude of the coefficients are affected by the increasingly unstable adjusting station queue and also the increasing number of replications used to compute the coefficients. These effects are portrayed in a series of illustrations that are used to contrast the information provided by the WR and WRi CV models. The second experiment provides a more specific analysis of these affects that incorporate the information provided by considering the statistical significance of the dependent variables of the regression models using all available replications.

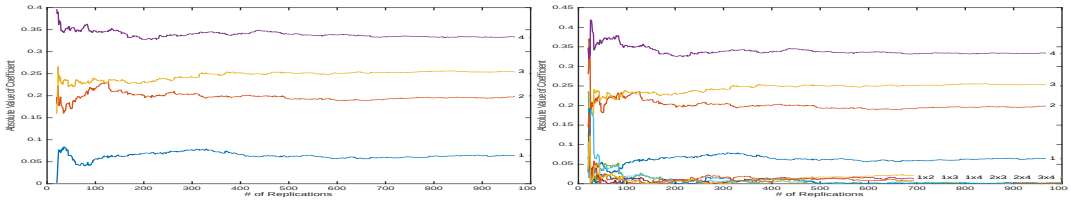
2.3.3.1 Experiment 1: Replicated Linearly Regressed Coefficients.

In this experiment the linear regressions for the WR and WRi CV models were evaluated, in an iterative fashion, as the number of replications was incremented by one additional replication from 20 to 1,000 total replications. This was accomplished for each of the ten configurations of the probability of failing inspection, $X_{p(FI)}$. The magnitude of the coefficients for these models provide us the information of which stochastic processes are producing the most variance in the model. This is a direct result of the CVs being independent from each other and also standardized. More precisely, the coefficient tells us the effect that a one standard deviation change in a CV has on the mean response. The absolute value of the estimating coefficients for each of the iterations are illustrated in Figures 2a - 2t, where the left side of the figure illustrates the scenarios where all available CVs excluding interactions, WR, is used and the right side of the figure displays the situations when all available CVs including interactions, WRi, is used. Hence, the viewpoint can be contrasted horizontally, that is provided by including unbiased second order interactions to the CV model that excludes them as we introduced increasing amounts of instability to the adjusting station queue.



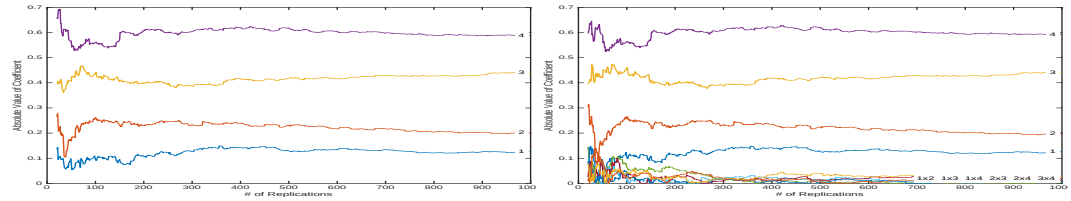
(a) $X_{p(FI)} = 0.05$, WR

(b) $X_{p(FI)} = 0.05$, WRi



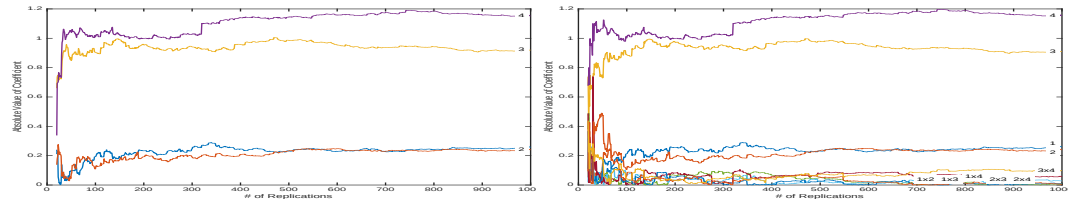
(c) $X_{p(FI)} = 0.10$, WR

(d) $X_{p(FI)} = 0.10$, WRi



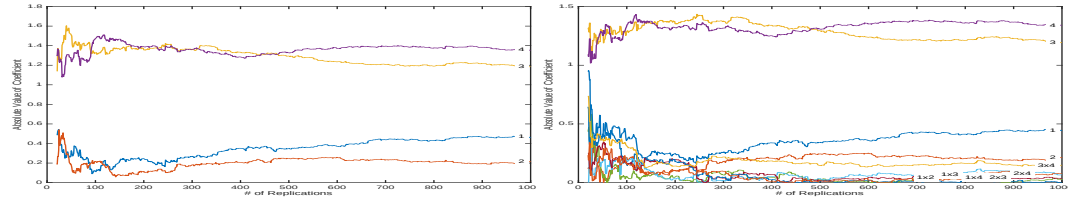
(e) $X_{p(FI)} = 0.15$, WR

(f) $X_{p(FI)} = 0.15$, WRi



(g) $X_{p(FI)} = 0.20$, WR

(h) $X_{p(FI)} = 0.20$, WRi

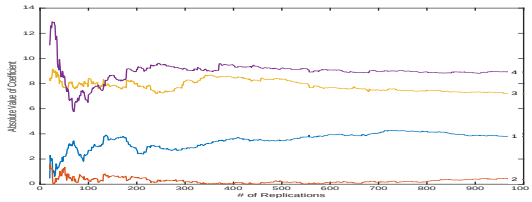


(i) $X_{p(FI)} = 0.25$, WR

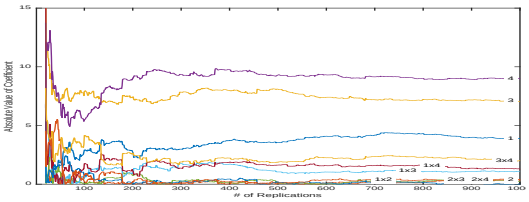
(j) $X_{p(FI)} = 0.25$, WRi

1) Interarrival 2) Inspector 3) Adjustor 4) Fail Inspection

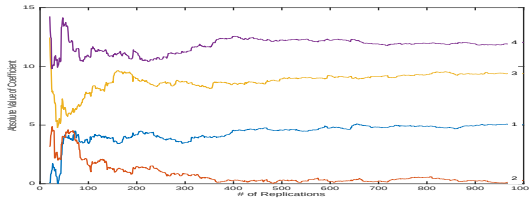
Figure 14. Absolute vale of estimating coefficient for 100-1,000 replications



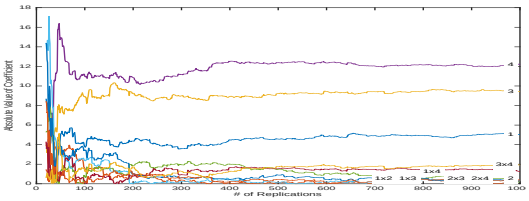
(k) $X_{p(FI)} = 0.30$, WR



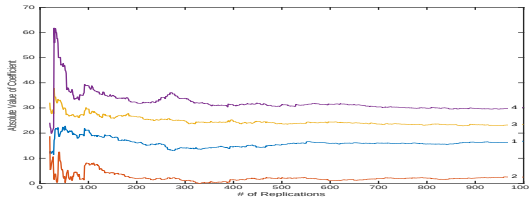
(l) $X_{p(FI)} = 0.30$, WRi



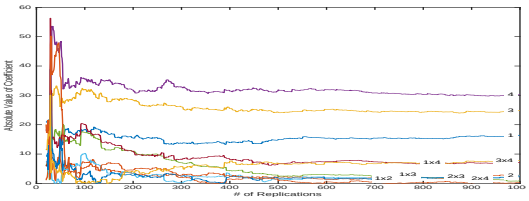
(m) $X_{p(FI)} = 0.31$, WR



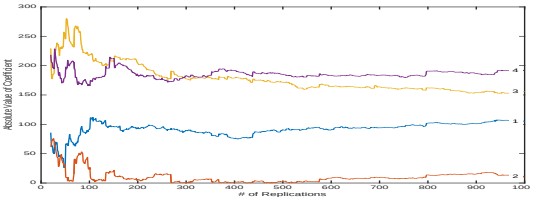
(n) $X_{p(FI)} = 0.31$, WRi



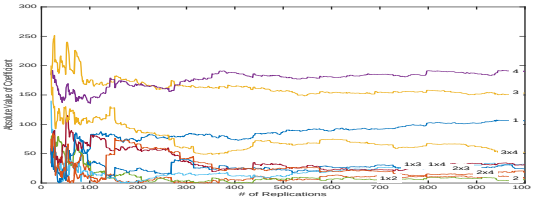
(o) $X_{p(FI)} = 0.32$, WR



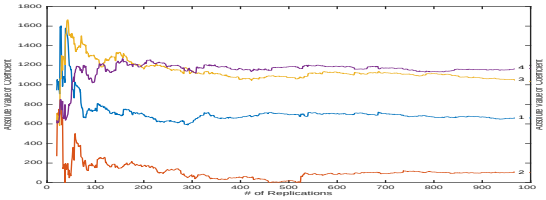
(p) $X_{p(FI)} = 0.32$, WRi



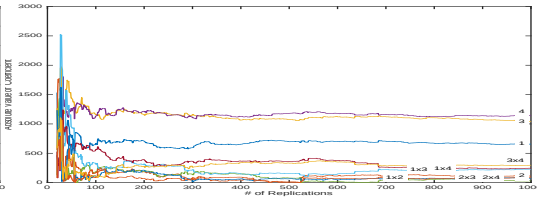
(q) $X_{p(FI)} = 0.33$, WR



(r) $X_{p(FI)} = 0.33$, WRi



(s) $X_{p(FI)} = 0.333$, WR



(t) $X_{p(FI)} = 0.333$, WRi

1) Interarrival 2) Inspector 3) Adjustor 4) Fail Inspection

Figure 2 (cont.): Absolute value of estimating coefficient for 100-1,000 replications

By contrasting vertically through Figures 2a - 2t for both CV models, wRi and WRi, the magnitude of the coefficients were observed for the first order dependent variables that can be linked with the adjusting station queue, $W_{\mu(A)}$ and $R_{p(FI)}$, increased more rapidly than the other first order dependent variables. Further, by contrasting horizontally it can be seen on the right side of Figures 2a - 2t for the CV interaction model WRi, show the magnitude of the coefficients for the second order interactions also increased as the failure rate, $X_{p(FI)}$ was increased. More specifically when $X_{p(FI)} \geq 0.30$, many of these second order interactions are greater in magnitude than the coefficient for the first order dependent variable for the inspection station, $W_{\mu(I)}$. This is a perspective that can not be inferred from the left side of Figures 2a - 2t, that depict the CV model that exclude the interactions, WR. Finally, these observed macro-level deductions are evident with sample sizes as low as 100 replications.

The heterogenous movements in the magnitude of the coefficients for the first order dependent variables and the second order interactions, suggest they are responding disparately to the changing system dynamics in the simulation as the adjusting station queue became more unstable. Both sides of Figures 2a - 2t demonstrate similar behaviors of the first order effects, however, only the right side of the figures illustrate the more subtle nonlinear reactions occurring with the second order interactions. Understanding these subtle trends gives a more robust viewpoint of the system dynamics, giving graphical support for the intuition.

2.3.3.2 Experiment 2: Linearly Regressed, Statistically Significant Coefficients w/ Max Replications.

The final experiment of this second part of the study, extracts from the first experiment the sets of coefficients for the CV models, WR and WRi, that are computed

using all 1,000 replications for each of the ten configurations of the failure rate, $X_{p(FI)}$. This coefficient information is summarized in Tables 7 and 8 (bold font are statistically significant $p < 0.05$), with the pertinent regression statistics (root mean squared error, coefficient of determination (R^2), and adjusted coefficient of determination (Adj. R^2)).

Table 7. Estimating coefficients for WR models using 1,000 replications

$X_{p(FI)} =$	.05	.10	.15	.20	.25
$R^2 =$	0.8664	0.8331	0.7408	0.7093	0.5714
Adj. $R^2 =$	0.8659	0.8324	0.7397	0.7082	0.5697
rMSE =	0.1106	0.2104	0.4523	0.9715	1.6216
0: Intercept	**8.187**	**10.355**	**13.537**	**18.729**	**29.493**
1: $\boldsymbol{W}_{\lambda}$	**0.031**	**0.064**	**0.126**	**0.257**	**0.464**
2: $\boldsymbol{W}_{\mu(I)}$	**0.184**	**0.199**	**0.202**	**0.239**	**0.212**
3: $\boldsymbol{W}_{\mu(A)}$	**0.122**	**0.255**	**0.440**	**0.919**	**1.185**
4: $\boldsymbol{R}_{\mathrm{p(FI)}}$	**0.176**	**0.332**	**0.589**	**1.153**	**1.354**

$X_{p(FI)} =$	.30	.31	.32	.33	.333
$R^2 =$	0.5026	0.5334	0.4955	0.3897	0.3231
Adj. $R^2 =$	0.5006	0.5315	0.4935	0.3872	0.3176
rMSE =	12.1374	15.3594	43.5201	332.9011	1784.7346
0: Intercept	**71.005**	**99.245**	**171.005**	**659.113**	**6633.920**
1: $\boldsymbol{W}_{\lambda}$	**3.759**	**5.029**	**16.733**	**105.104**	**663.182**
2: $\boldsymbol{W}_{\mu(I)}$	0.365	0.288	2.456	10.061	103.392
3: $\boldsymbol{W}_{\mu(A)}$	**7.195**	**9.381**	**23.503**	**156.14**	**1029.921**
4: $\boldsymbol{R}_{\mathrm{p(FI)}}$	**8.865**	**12.002**	**30.067**	**191.286**	**1162.255**

Table 8. Estimating coefficients for WRi models using 1,000 replications

$X_{p(FI)} =$	.05	.10	.15	.20	.25
$R^2 =$	0.8679	0.8348	0.7427	0.7137	0.5773
Adj $R^2 =$	0.8666	0.8331	0.7401	0.7108	0.5731
rMSE =	0.1103	0.2100	0.4520	0.9672	1.6153
0: Intercept	**8.189**	**10.356**	**13.537**	**18.729**	**29.498**
1: $\boldsymbol{W}_{\lambda}$	**0.032**	**0.064**	**0.127**	**0.259**	**0.446**
2: $\boldsymbol{W}_{\mu(I)}$	**0.185**	**0.200**	**0.198**	**0.239**	**0.209**
3: $\boldsymbol{W}_{\mu(A)}$	**0.122**	**0.254**	**0.441**	**0.911**	**1.193**
4: $\boldsymbol{R}_{\mathrm{p(FI)}}$	**0.176**	**0.332**	**0.592**	**1.155**	**1.340**
1 x 2	0.007	0.003	0.004	0.012	0.024
1 x 3	0.000	0.000	0.003	0.028	0.067
1 x 4	0.004	0.012	0.017	0.057	0.033
2 x 3	0.001	0.002	0.012	0.009	0.003
2 x 4	0.006	0.000	0.012	0.006	0.077
3 x 4	**0.008**	**0.016**	**0.027**	**0.0978**	**0.148**

Table 8 (cont.): Estimating coefficients for WRi models using 1,000 replications

$X_{p(FI)}$ =	.30	.31	.32	.33	.333
R^2 =	0.5268	0.5456	0.5304	0.4204	0.3799
Adj R^2 =	0.5220	0.5410	0.5256	0.4146	0.3697
rMSE =	11.8736	15.2022	42.1181	325.3840	1642.4894
0: Intercept	**70.892**	**99.217**	**170.402**	**659.994**	**6631.741**
1: $\boldsymbol{W}_{\lambda}$	**3.872**	**5.027**	**16.342**	**105.751**	**658.234**
2: $\boldsymbol{W}_{\mu(I)}$	0.390	0.240	2.713	8.003	111.825
3: $\boldsymbol{W}_{\mu(A)}$	**7.079**	**9.409**	**24.775**	**153.39**	**1040.937**
4: $\boldsymbol{R}_{\mathrm{p(FI)}}$	**8.948**	**12.159**	**30.154**	**189.630**	**1140.466**
1 x 2	0.229	0.211	0.884	6.159	30.546
1 x 3	**1.119**	**1.301**	**2.804**	**31.957**	**225.090**
1 x 4	**1.264**	**1.258**	**7.175**	**31.326**	**244.797**
2 x 3	0.031	0.300	2.288	24.071	75.992
2 x 4	0.312	0.380	0.253	18.126	86.316
3 x 4	**2.026**	**1.902**	**7.771**	**52.359**	**299.494**

Its noted from Table 7, as the probability of failing inspection, $X_{p(FI)}$, is increased the introduction of the statistically significant ($p < 0.05$) second order interaction terms (marked by bold font) point to the increasing nonlinear effects caused by the feedback loop. The important second order interactions can be identified, (i.e., 1×3, 1×4, 2×3, and 3×4), as they manifest and accumulate around any components that are associated with failing inspection, $R_{p(FI)}$, or queued for adjustment, $W_{\mu(A)}$. Its also noted, the magnitude of the coefficients for these second order interactions are 300+% greater than the coefficient of the first order dependent variable for the in-

spection station, $W_{\mu(I)}$. Finally, all regression metrics for the WRi model are superior to the WR model, for every configuration of $X_{p(FI)}$.

2.4 Conclusions

Internalizing second order interactions provided much greater visibility of the subtle changes of the system dynamics as additional system variance was induced incrementally across ten configurations. The statistical framework was provided for the introduction of these second order interactions to the original CV formulation which retains the ability to produce an unbiased estimate for the simulation response of interest. A two-part study provided the results that confirmed the intuition provided by this new statistical framework.

The first part of the study showed that much of the latent variance reduction can be retained with these additional interaction estimators, when provided sufficient replications to offset the influence of the t-statistic on the confidence interval. In some cases the variance reduction improved with the introduction of the second-order CV interactions.

The second part of the study demonstrated the regression model that included second order CV interactions terms outperformed the model that excluded these new terms. This can also be graphically observed from the second order CV model, as a more robust understanding of how the relationships between the stochastic processes were reacting to the changing system dynamics of the simulation.

The totality of the empirical results from these two studies provided a much better understanding of the system dynamics of the simulation that suggests when employed in a much bigger scheme could improve the simulation practitioners inference of valuable insight from the system.

III. Improved Multi-Population Least Squares Regression Metamodels for Simulations Using Linear Control Variates with Interactions

3.1 Introduction

In conventional terms, a metamodel is a model of an underlying model. The interest here is in computer simulation which, in of itself, is a model that acts as a representation of a real world system. Depending on the scale of the simulation or the complexity of its internal logical components, directly analyzing the simulation can be computationally intensive for the former case, if not entirely intractable in the latter case. Provided sufficient precision, a metamodel can be used to conduct sensitivity analysis (see Barton [9]), estimate unknown mean response surfaces (see Sargent[111]), or make available indirect simulation optimization (see Barton [11] and Andradottir [6]). In general, a lower resolution metamodel can act as an analytic proxy for a higher resolution simulation. Relevant surveys of this field of study include, Barton [11] and Friedman [36].

Variance reduction techniques (VRTs) have been developed in parallel of metamodels, to further improve the efficiency of simulation based experiments. Our interest in this paper limits the context of these techniques to control variates (CVs) (see Wilson [128, 129] for broader surveys of VRTs). Lavenberg [69, 68] developed CVs for single response simulations with multiple CVs, also developing two efficiency measures, the minimum variance ratio and loss factor to account for variance reduction or inflation as CVs are selected for inclusion. Rubinstein [109] extended this development for multiple response simulations. Venktraman [122] provides a simpler formulation for the variance ratio and loss factor given by Rubinstein [109].

Nozari [92] unified the efficiency mechanisms of general linear regression metamodels and CVs into a single formulation for single response simulations, also providing efficiency measures analogous to the CVs for single population context. Porta Nova [100] generalized this formulation for multiple response simulations. Tew [120, 121] also extended the formulation by Nozari [92] for a single response simulation with multiple control variates by combining all correlation-based variance reduction techniques into a single metamodel.

Two subsequent articles examine new types of CVs that induce properties favorable to the context of this paper. Wilson & Pritsker [132] provide analysis for what they call "work" variables that are characterized as asymptotically standardized CVs. Bauer & Wilson [16] introduced "routing" CVs that are evaluated from the stochastic branching processes commonly found in simulations that are similarly characterized as asymptotically standardized CVs. These two types of CVs treat a large proportion of the stochastic processes found in most simulations.

The organization for this chapter is as follows. Section 3.2 will provide the extended statistical framework that is woven from several of the articles mentioned from the literature review in Chapter I. Section 3.3 will describe the experimental process for the case study. Section 3.4 presents the findings of the case study in the prior section. Section 3.5 will summarize and conclude this chapter.

3.2 Second Order Interaction Metamodel with Multiple Control Variates for a Multiple Population, Single Response Simulation

The first order least squares regression metamodel with first order control variates in equation (1.2.3.31) can be extended to the second order least squares regression metamodel with first order control variates. Let $\beta = [\beta_1, \beta_2, \ldots, \beta_{(M)}]$ be a $1 \times M$ row vector of linear estimating coefficients for the $M \times 1$ column vector of designed input

variables, X, and B be an $M \times M$ matrix of second order interaction and quadratic estimating coefficients, also for the $M \times 1$ column vector of designed input variables, X, that is given as

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & \cdots & \Delta_{1,M} \\ B_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ B_{M,1} & \cdots & \cdots & \Delta_{M,M} \end{bmatrix}. \tag{3.2.0.1}$$

Now let $W = [W_1, W_2, \ldots, W_Q]'$ be a $Q \times 1$ column vector of standardized "work" CVs and $R = [R_1, R_2, \ldots, R_P]'$ be a $P \times 1$ column vector of standardized "routing" CVs, then we can define $Z = [W_1, W_2, \ldots, W_Q, R_1, R_2, \ldots, R_P]'$ as a $(Q + P) \times 1$ column vector of vertically concatenated standardized "work" and "routing" CVs. Let $\delta = [\delta_1, \delta_2, \ldots, \delta_{(Q+P)}]$ be a $1 \times (Q+P)$ row vector of linear estimating coefficients for the $(Q + P) \times 1$ column vector of vertically concatenated standardized "work" and "routing" CVs, Z. We can now define the second order least squares regression metamodel with first order control variates as

$$y_i = \beta \cdot X_i + \delta \cdot Z_i + X_i' \cdot B \cdot X_i + e_i, \tag{3.2.0.2}$$

where y_i is the i^{th} observed simulation response, X_i is the i^{th} observed $1 \times (M + 1)$ row vector of deterministic input variates (with a leading 1 for the intercept term).

To extend the second order least squares regression metamodel with first order control variates from equation (3.2.0.2), to an unbiased second order least squares regression metamodel with second order control variate interactions, we let Δ be a $(Q + P) \times (Q + P)$ matrix of second order interaction and quadratic estimating

coefficients for the $(P+Q)\times 1$ column vector of standardized "work" and "routing" CVs, Z, given as

$$\Delta = \begin{bmatrix} \Delta_{1,1} & \Delta_{1,2} & \cdots & \Delta_{1,(Q+P)} \\ \Delta_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Delta_{(Q+P),1} & \cdots & \cdots & \Delta_{(Q+P),(Q+P)} \end{bmatrix}, \tag{3.2.0.3}$$

and let Γ be a $(Q+P)\times M$ matrix of second order interaction estimating coefficients between the $M\times 1$ column vector of designed input variables, X, and the $(Q+P)\times 1$ column vector of standardized "work" and "routing" CVs in Z, given as

$$\Gamma = \begin{bmatrix} \Gamma_{1,1} & \Gamma_{1,2} & \cdots & \Gamma_{1,M} \\ \Gamma_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \Gamma_{(Q+P),1} & \cdots & \cdots & \Gamma_{(Q+P),M} \end{bmatrix}. \tag{3.2.0.4}$$

By appending these two new sets of second order interaction and quadratic estimating coefficients, Δ and Γ, respectively, to equation (1.2.3.31), gives us

$$y_i = \beta \cdot X_i + \delta \cdot Z_i + X_i' \cdot B \cdot X_i + Z_i' \cdot \Delta \cdot Z_i + Z_i' \cdot \Gamma \cdot X_i + e_i, \tag{3.2.0.5}$$

As was observed in the previous chapter on the second order control variates for single response single population simulations, this formulation is affected by the bias associated with introducing pure quadratic formulations of standardized control

variates, i.e., $E\{Z' \cdot \Delta \cdot Z\} = \text{tr}\{\Delta\}$. For the same reason, we define $\tilde{\Delta} = \Delta - \text{diag}\{\Delta\}$, that is, place zeros as the estimating coefficient for all second order quadratic effects associated with $\text{diag}\{Z'Z\}$ then we can substitute $\tilde{\Delta}$ for Δ into equation (3.2.0.5), then the least squares regression model with second order control variate interactions becomes

$$y_i = \beta \cdot X_i + \delta \cdot Z_i + X_i' \cdot B \cdot X_i + Z_i' \cdot \tilde{\Delta} \cdot Z_i + Z_i' \cdot \Gamma \cdot X_i + e_i, \tag{3.2.0.6}$$

and is now an unbiased estimate for y.

3.3 Case Study

This section provides analysis from the study that determined the least squares regression model with second order control variate interactions, by and large, outperforms the second order least squares regression model that excludes second order control variate interactions. An effort is also made to demonstrate the inclusion of these new terms provides a better understanding of the system dynamics in the components of the simulation that are contributing a significant impact to its output.

3.3.1 Simulation Analysis.

Before conducting experiments of the simulation (see Section 1.2.6 for review of simulation) and analyzing results, basic Jackson formulas for open networks (see Section 1.2.5) were used to evaluate the simulation layout, its components, and respective input settings which provided the true long run average sojourn time of the televisions and the true long run utilization of the server for the adjustor queue that is modeled in the simulation. From these Jackson formulas a feasible region was derived that was characterized by a significant degree of variance across the entire designed

area while maintaining stable queues, more precisely, when utilization of the queues does not exceed a ratio above 1. The range of these input settings and outputs are summarized in Tables 9 and 10 below, or see Tables 17 - 20 in Appendix A.A.1 for specific information for each individual designed setting.

Table 9. Range of Simulation Input Settings and Output Ranges

Label	Input/Output Description	Minimum	Maximum
X_{λ}	TV mean inter-arrival time	16.25	17.25
$X_{\mu(I)}$	inspection mean service time	19.5	21.5
$X_{\mu(A)}$	adjustor mean service time	28	32
$X_{p(FI)}$	probability of failing inspection	0.325	0.345
Y	*True* average TV sojourn time in system	219.64	949.22
U_A	*True* Adjustor Utilization	0.8402	0.9679
$\bar{Y}$	*Observed* average TV sojourn time in system	163.67	3307.60
$\bar{U}_A$	*Observed* Adjustor Utilization	0.7814	0.99642

Reviewing Table 9, note the tremendous amount of variance that is observed in the responses with relation to the true range of the responses evaluated from Jackson formulas. For example, the range of the true average TV sojourn time in system is 219.64-949.22, however the range observed (across 250 replications) in the simulation is 163.67-3307.60. This can be attributed to the variability that is induced from the adjustor utilization rate that is, $\rho \geq\sim 0.85$. We saw in chapter II, (see Table 2) the effects that manifest on the response for this simulated system when a single queue is exceptionally close to instability ($\rho \approx 0.999$). Despite this variability, the observed long run mean responses, $\bar{Y}$ and $\bar{U}_A$, were relatively close to the true long run mean responses, Y and U_A, (see Tables 17 - 20 in Appendix A.A.1), at all 73 design points. The largest outlying difference between the two surfaces occurred at design point ID#

62, from Tables 17 - 20 in Appendix A.A.1. At this setting (a duplicate point for FF, CCc, & CCf), the utilization rates for both of the inspector and adjustor stations were close to instability, $rho \geq 0.90$.

Furthermore, of the 73 design points, two design points (ID# 31 & 62) observed greater than 5%, but less than 6%, difference between the observed mean and the true mean. One design point (ID# 11) was greater than 3%, but less than 4%, difference between the observed mean and the true mean. Four design points (ID# 4, 26, 46, & 75) was greater than 2%, but less than 3%, difference between the observed mean and the true mean. The remaining 66 design points observed strictly less than 2% difference between the observed mean and the true mean. This suggests the selected run length, 500K time units, for the entire set of designed points and across all five designed experiments, was sufficient in length for the far majority of the total design points.

Table 10. Description of Simulation Control Variates

Label	CV Description	Distribution
W_{λ}	"work" CV for TV inter-arrival time	$\backsim N(0,1)$
$W_{\mu(I)}$	"work" CV for inspection service time	$\backsim N(0,1)$
$W_{\mu(A)}$	"work" CV for adjustor service time	$\backsim N(0,1)$
$R_{p(FI)}$	"routing" CV for failing inspection	$\backsim N(0,1)$

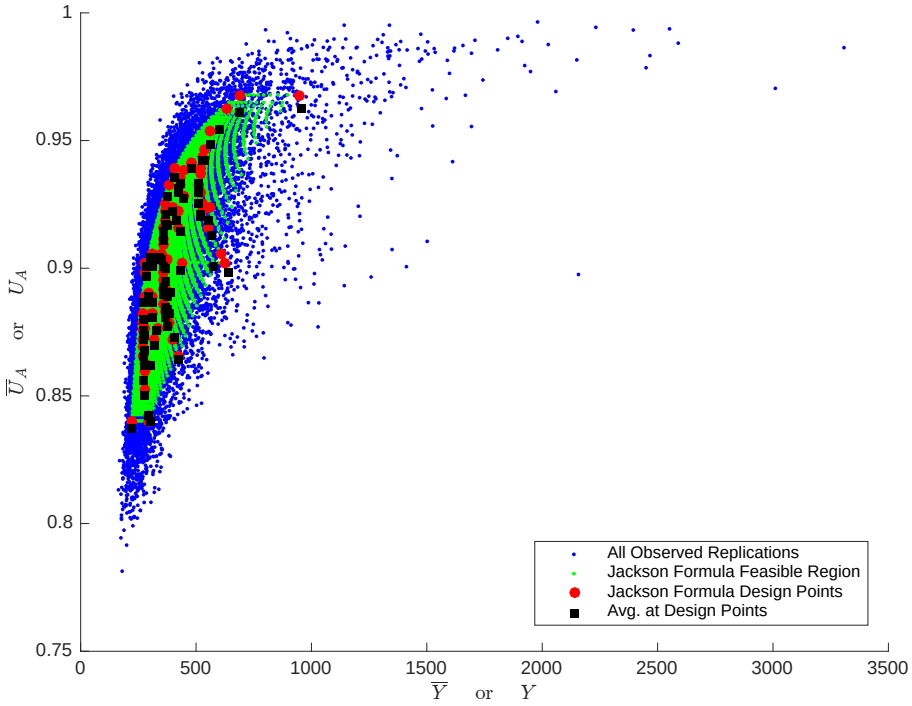
Five different designed experiments, summarized below in Table 11, were selected to conduct on the identified feasible region of the simulation that is reviewed in Section 1.2.6. There are 107 total design points across all five experiments. Of the 107, 34 points were duplicate design settings across multiple designed experiments. For these cases, only a single set of 250 replications were ran in the simulation. In the meta-model analysis, these sets of replications were re-used dependent on which

designed experiment was being analyzed. Eliminating the duplicative design points, reduced the total set of unique design points to 73 unique points that were run in the simulation. See Tables 17-20 in Appendix A.A.1 for the complete set of 73 design points that includes the information about their associated relationship(s) to the five types of designed experiments, their coded values, and the true (Jackson formulas) and observed (across 250 replications) long run mean responses for average TV sojourn time and adjustor station utilization for each design point.

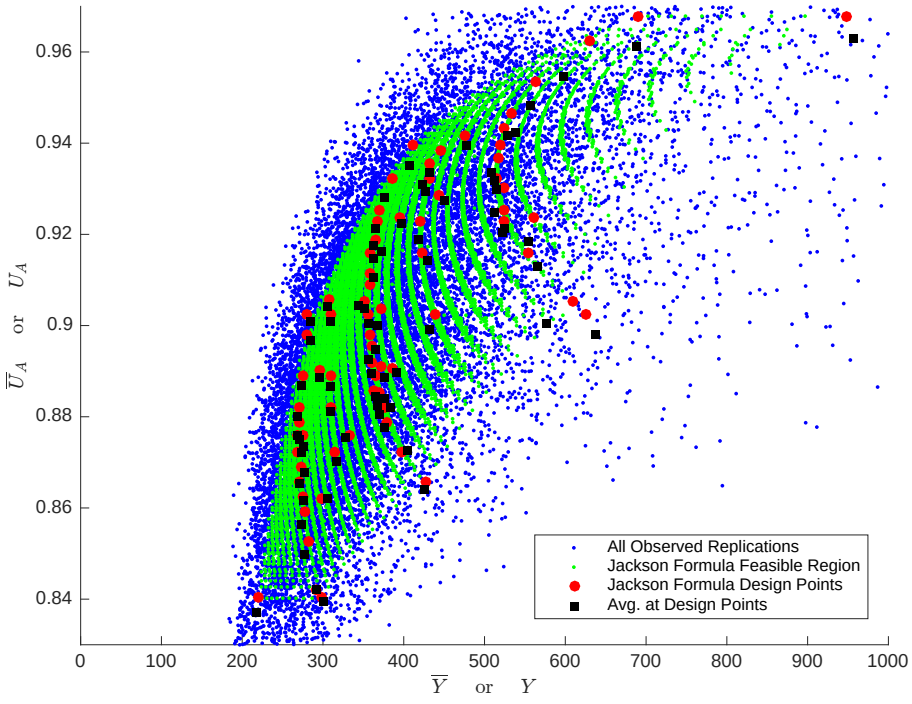
Each of the 73 points are replicated 250 times for 500,000 time units at each replication. The tremendously long run time that was selected, 500K time units, was to compensate for the inherent variability that was spoken of earlier (also see Figure 31). We saw in Chapter II, (see Table 2) when the utilization rate for the adjustor was near instability, $\rho_{\mu(A)} \geq 0.85$, the necessary run lengths required to observe values for the long run average TV sojourn time, $\widehat{W}_{G\&B}$, that were relatively close to the true long run average TV sojourn TV sojourn time, $W_{Jackson}$, required 300K+ time lengths, even up to 1.5 to 700 million time units when $\rho_{\mu(A)} = \{0.985, 0.999\}$, respectively. However, the latter two run-lengths would have required far too much computationally resources and time to conduct across 73 experiments. It was determined that replicating the run length for 500K time units, provided sufficient accuracy to the true long run response values, while demanding reasonable computational resources and time.

Further, to compensate for any transient bias, a truncation point was established so the first 50% of the time domain for each replication was eliminated. The simulation outputs, $\bar{Y}$ and $\bar{U}_{Ai}$, the three standardized "work" control variates W_{λ}, $W_{\mu(I)}$, $W_{\mu(A)}$, and the single standardized "routing" control variates $R_{p(FI)}$, (summarized in Table 10) were evaluated and recorded on only the last 250,000 time units of each replication of the simulation.

Figures 14a and 14b, illustrate the projection that occurs when the simulation or Jackson formulas transform a four dimensional input into two dimensional output. The red circles represent the Jackson formula projection of the four dimensional input space of X to the two dimensional output space made up of Y and U_{Ai}. Similarly, the blue dots represent the simulation based projection from X to $\bar{Y}$ and $\bar{U}_A$. The black squares represent the average of a design setting's output across its 250 replications, $\overline{\overline{Y}}$ and $\overline{\overline{U}}_A$. Finally, the green dots represent the Jackson formula projection of a fine meshed grid of points (14,529 total points) between the values of the input settings for X, from Table 9, which also represents the *true* mean surface feasible region.



(u) Low Resolution View of all 73 Design Points



(v) High Resolution View of all 73 Design Points

Figure 14. Simulation / Jackson Formula Projections for All 73 Design Points

Furthermore, Figures 14a and 14b demonstrate the exceptional amount variance, mentioned earlier, that is observed from the simulation (blue dots). However, despite this variance, the relative distance between the true long run responses (red dots) and the average observed long run responses (black dots) is very close, which gives more support to the selected run length, 500K time units, and the number of replications, 250. Also see Figures 31a - 31d in Appendix A.A.3 that illustrate the same type of figures that have the context scoped specifically for each of the five designed experiments that are summarized below in Table 11.

Table 11. Description of Designed Experiments

Label	Type of Design	# of Settings
FF	2^4 Full Factorial	16
BB	Box-Behnken	25
LH	Latin Hypercube	16
CCf	Central Composite - Faced	25
CCc	Central Composite - Circumscribed	25

Also see Tables 17 - 20 in Appendix A.A.1 as an extension to Table 11.

3.3.2 Non-linear Transformation of Simulation Responses.

The objective for the research in this chapter is to showcase the superior utility of the second order interactions of the standardized "work" and "routing" variates, in a scenario that would prove difficult for the standard formulations of least squares regression meta-models that have excluded these types of interaction terms based on what is available in the literature. In many ways, the simulation would have a reasonable degree of non-linearity inherent with just the existing single feedback loop,

and also with the intent of selecting a designed region that would exhibit an extreme amount of system variance due to queues being near instability.

As an effort to further demonstrate the usefulness of these types of interaction terms, an additional layer of complexity was induced into the methodology. The two output parameters from the simulation were passed through four different non-linear projections, $\bar{Y}$ and $\bar{U}_{Ai}$ to a univariate "cost function" response. Equations (3.3.2.8) - (3.3.2.11) and illustrated in Figures 17a-17d. These transformations act as "cost functions" for each of the individual simulation design points.

The first step in this process was to code the values of the two simulation responses, $\bar{Y}$ and $\bar{U}_A$, to a range between 0 and 1,

$$\bar{Y}^{coded} = \frac{\bar{Y} - 1750}{3500} + \frac{1}{2} \tag{3.3.2.1}$$

$$\bar{U}_A^{coded} = \frac{\bar{U}_A - .875}{.25} + \frac{1}{2}. \tag{3.3.2.2}$$

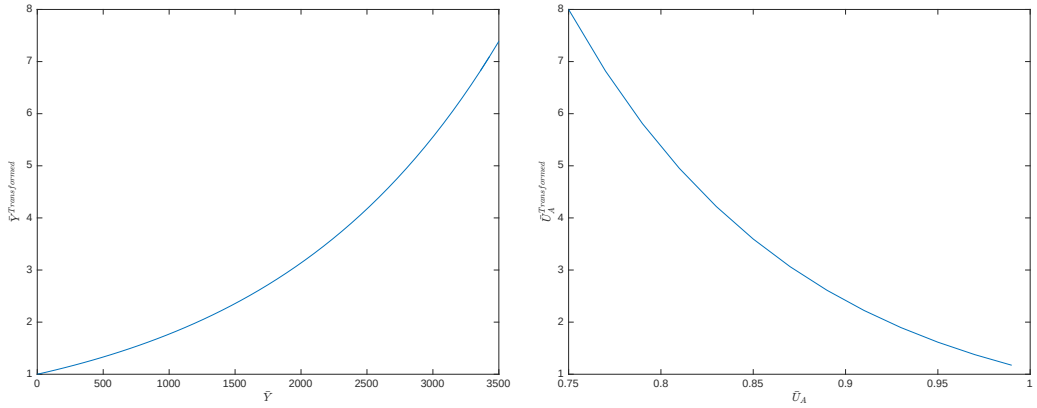
At their natural settings, $\bar{Y}$ and $\bar{U}_A$, are disparately scaled in range, which coding resolves. In the next step, the coded responses for, $\bar{Y}$ and $\bar{U}_A$, were passed through a non-linear mathematical transformation, given as

$$tf\bar{Y} = \exp^{(2 \cdot \bar{Y}^{coded})} \tag{3.3.2.3}$$

$$tf\bar{U}_A = \exp^{(-2 \cdot \bar{U}_A^{coded})}. \tag{3.3.2.4}$$

Figures 15a and 15b illustrate the relationship between the responses, $\bar{Y}$ and $\bar{U}_A$, with their non-linear transformed values, $\bar{Y}^{transformed}$ and $\bar{U}_A^{transformed}$. It can be seen in these figures, that after coding the responses, before the non-linear transformation, re-scaled them to a point that created a better balance in the effort to define a

cost function that depended on opposing slopes of the two transformed and coded responses.



(a) Plot of $tf\bar{Y}$ (b) Plot of $tf\bar{U}_A$

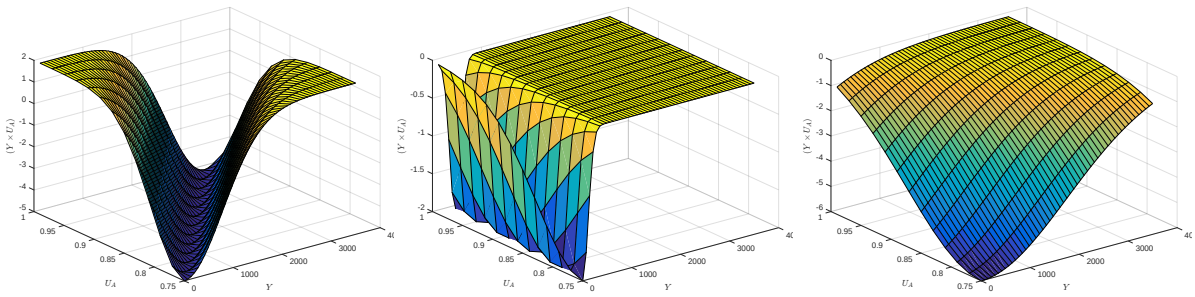
Figure 15. Non-linear transformations of Y^{coded} and U_A^{coded}

The third step in this endeavor was to develop a non-linearly transformed interaction term between the two responses, $\bar{Y}$ and $\bar{U}_A$, to provide sufficient complexity to this exercise. Figure 16, illustrates these interactions in equations (3.3.2.5) - (3.3.2.7).

$$tfYU_1(\bar{Y}^{coded}, \bar{U}_A^{coded}) = -7 \cdot \exp\Big(-10 \cdot (\bar{Y}^{coded} - \bar{U}_A^{coded})^2\Big) \tag{3.3.2.5}$$

$$tfYU_2(\bar{Y}^{coded}, \bar{U}_A^{coded}) = -2 \cdot \exp\Big(-5 \cdot (\bar{Y}^{coded} - 10 \cdot \bar{U}_A^{coded})^2\Big) \tag{3.3.2.6}$$

$$tfYU_3(\bar{Y}^{coded}, \bar{U}_A^{coded}) = -6 \cdot \exp\Big(-2 \cdot (\bar{Y}^{coded} + \cdot \bar{U}_A^{coded})^2\Big) \tag{3.3.2.7}$$



(a) Plot of $tfYU_1$ (b) Plot of $tfYU_2$ (c) Plot of $tfYU_3$

Figure 16. Non-linear transformations of interaction terms between Y^{coded} and U_A^{coded}

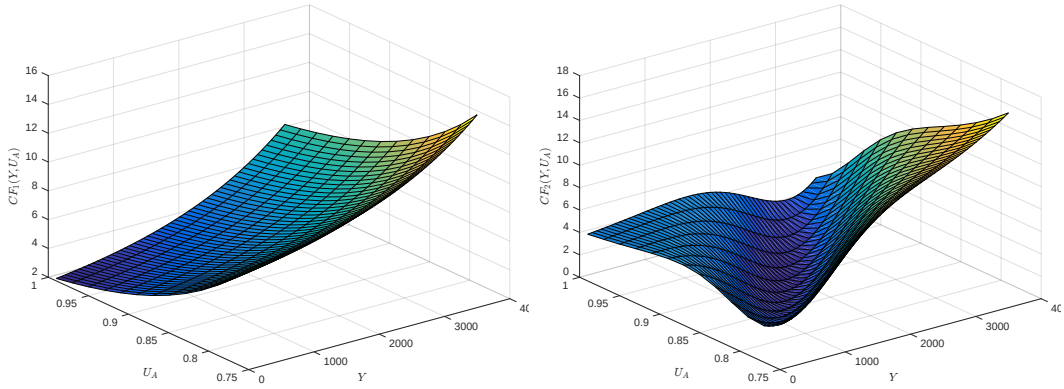
The final step in this effort was to formulate four complete non-linear transformations that combined the non-linear transformations for each of the individual responses, $tf\bar{Y}$ and $tf\bar{U}_A$, with the three non-linear transformation of the interaction between coded responses, $tfYU_1$, $tfYU_2$, and $tfYU_3$.

$$CF_1 = tf\bar{Y} + tf\bar{U}_A \tag{3.3.2.8}$$

$$CF_2 = tf\bar{Y} + tf\bar{U}_A + tf\overline{YU}_1 \tag{3.3.2.9}$$

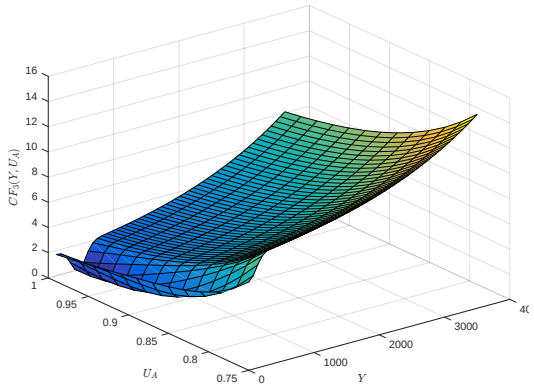
$$CF_3 = tf\bar{Y} + tf\bar{U}_A + tf\overline{YU}_2 \tag{3.3.2.10}$$

$$CF_4 = tf\bar{Y} + tf\bar{U}_A + tf\overline{YU}_3 \tag{3.3.2.11}$$

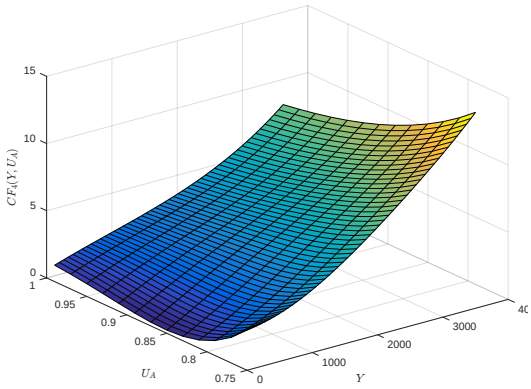


(a) $CF_1(\bar{Y}, \bar{U}_A)$

(b) $CF_2(\bar{Y}, \bar{U}_A)$



(c) $CF_3(\bar{Y}, \bar{U}_A)$



(d) $CF_4(\bar{Y}, \bar{U}_A)$

Figure 17. Surface Plots of Four Nonlinear Cost Function Transformations

3.3.3 Meta-Experimentation for Least Squares Regression Meta-models.

A meta-experiment was conducted on the data that was accumulated from the simulation across all the design settings associated with a particular designed experiment. For each designed experiment, various characteristics of the simulation data were varied to inspect if any significant affects are evident. These varied components of the meta-experiment are:

- # of K folds = {25, 20, 17, 14, 13, 11, 10}

- selected designed experiment
  - FF: 2^4 Full Factorial
  - CCf: Central Composite - Faced
  - CCc: Central Composite - Circumscribed
  - BB: Box-Behnken
  - LH: Latin Hypercube
- type of output or nonlinearly transformed composition of outputs
  - Y: Average TV Sojourn Time
  - U_A: Adjustor Server Utilization
  - CF_1: Cost Function 1
  - CF_2: Cost Function 2
  - CF_3: Cost Function 3
  - CF_4: Cost Function 4
- type of least squares (LS) regression metamodel invoked
  - $1LSR_{noCV}$: first order LS Regression Metamodel *excluding* CVs
  - $1LSR_{CV}$: first order LS Regression Metamodel *including* CVs
  - $2LSR_{noCV}$: second order LS Regression Metamodel *excluding* second order CV interactions
  - $2LSR_{CV}$: second order LS Regression Metamodel *including* second order CV interactions

Figure 18 illustrates the steps of the meta-experiment that was conducted as part of the study in this chapter. The first step, illustrated in the top bar of Figure 18,

was the selection of one of the five designed experiments, $\{FF,\ CCf,\ CCc,\ BB,\ LH\}$, one of the six responses, $\{Y,\ U_A,\ CF_1,\ CF_2,\ CF_3,\ CF_4\}$, and one of the cross-validation sets of K folds, $\{25,\ 20,\ 17,\ 14,\ 13,\ 11,\ 10\}$. These folds also represent the equal distribution of a percent of data points from each design point into one of the K folds. For example, if the selected size of the cross-validation set is 13, then $\lfloor 250/13 \rfloor = 19$ replications (when rounded down) from each of the design points of a designed experiment is placed into a single fold, or approximately $19/250 \approx 8\%$ of the data from each design point. In this example, if the selected design was the 2^4 full factorial, FF, with 16 design points (see Tables 17 - 20 in Appendix A.A.1), then each kth fold would receive $16 \cdot 19 = 304$ total data points. Table 12 below provides an overview of this method across all the sets of cross-validation folds. The selected number of folds in this study were selected, because in practice, many times simulation practitioners have the resources to run only a small amount (10 - 25) of simulation replications, which became the minimum and maximum of the size of folds in this study. Furthermore, limiting the size of the folds also serves the purpose of limiting the information that the meta-models can use to train, making predictions more difficult. A reasonable number of fold values between the selected minimum and maximum was also selected for completeness of the study.

Table 12. Description of K fold method

# of folds K	# of replications per design point N	# of points per fold $\lfloor 250/K \rfloor$	% of available data per fold $\lfloor 250/K \rfloor / N$
25	250	10	4%
20	250	12	$4.8\% \approx 5\%$
17	250	14	$5.6\% \approx 6\%$
14	250	17	$6.8\% \approx 7\%$
13	250	19	$7.6\% \approx 8\%$
11	250	22	$8.8\% \approx 9\%$
10	250	25	10%

The second step, after the selection of the type of design, response, and number of folds, is the even distribution of data which is illustrated in Figure 18, directly below the top orange bar. For example, in this step a purple dot from each of the design points are all selected from the same relative position in each design point and placed into a single fold that receives only purple dots. This step iteratively cycles through all K folds, until all data from each design point of the simulation is distributed out to K independent cross-validation folds.

Continuing the example of 13 folds, each purple dot represents 19 observations from a given design point. The collection of all the purple dots constitutes a fold. All subsequent folds are constructed in a similar fashion.

The third step, illustrated in the two grey bars of Figure 18, encompasses three steps:

- the selection of one of the four least square regression meta-models, $\{1LSR_{noCV}$, $1LSR_{CV}$, $2LSR_{noCV}$, $2LSR_{CV}\}$,

- training selected regression meta-model on the k-th cross-validation set of data, and
- invoking interior-point minimization routine on the trained regression meta-model.

This third step is accomplished for all regression meta-model formulations and through all K cross-validation folds, 6 statistics, θ_k, are recorded for subsequent analysis:

- $\hat{\theta}^*$: Predicted minimum of selected response
- $\theta^{\text{abs.diff}} = |\hat{\theta}^* - \theta^*|$: Absolute difference between predicted minimum and the true minimum of a selected response
- $\theta^{\text{Norm}} = \text{Norm}\{\hat{\theta}^*_{\text{min}} - \theta^*_{\text{min}}\}$: L2 norm distance between the four dimensional coordinates, $\{X_\lambda, X_{\mu(I)}, X_{\mu(A)}, X_{p(FI)}\}$, for the predicted minimum and the true minimum for a selected response
- θ^{MSE}: mean squared error for the selected regression meta-model
- θ^{R^2}: coefficient of determination for the selected regression meta-model
- $\theta^{\text{Adj.}R^2}$: adjusted coefficient of determination for the selected regression meta-model

The fourth step, illustrated in the bottom green bar of Figure 18, simply calculates the average for each of the 6 statistics, $\overline{\theta}_i$, that were recorded in the prior step, across the K independent cross-validation folds, given in equations (3.3.3.1)-(3.3.3.6). Further, these statistics are evaluated for each of the four regression meta-models. The fourth step concludes by iteratively cycling 100 times back to the second step, to conduct a new random K-fold cross-validation of the same designed experiment,

response and number of K cross-validation folds. This step eventually collects 100 bootstrapped statistics, $\overline{\theta}_i$, that are averaged across K folds. This calculation for the ith bootstrapped iteration is shown in the equations directly below.

$$\bar{\theta}_i^{*} = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}_k^{*} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.1}$$

$$\bar{\theta}_i^{\text{abs.diff}} = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}_k^{\text{abs.diff}} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.2}$$

$$\bar{\theta}_i^{\text{Norm}} = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}_k^{\text{Norm}} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.3}$$

$$\bar{\theta}_i^{\text{MSE}} = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}_k^{\text{MSE}} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.4}$$

$$\bar{\theta}_i^{\text{R}^2} = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}_k^{\text{R}^2} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.5}$$

$$\bar{\theta}_i^{\text{Adj.}R^2} = \frac{1}{K} \cdot \sum_{i=1}^{K} \hat{\theta}_k^{\text{Adj. } R^2} \quad \text{for } i = 1, 2, \cdots 100 \tag{3.3.3.6}$$

The fifth and final step in this experiment, illustrated in the bottom orange bar of Figure 18, calculates a second average of the statistics for, $\overline{\overline{\theta}}_j$, across the 100 bootstrapped iterations of the previous step, for each of the four types of LS regression meta-models, j.

$$\overline{\overline{\theta}}_j^{*} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{*} \tag{3.3.3.7}$$

$$\overline{\overline{\theta}}_j^{\text{abs.diff}} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{abs.diff}} \tag{3.3.3.8}$$

$$\overline{\overline{\theta}}_j^{\text{Norm}} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{Norm}} \tag{3.3.3.9}$$

$$\overline{\overline{\theta}}_j^{\text{MSE}} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{MSE}} \tag{3.3.3.10}$$

$$\overline{\overline{\theta}}_j^{\text{R}^2} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{R}^2} \tag{3.3.3.11}$$

$$\overline{\overline{\theta}}_j^{\text{Adj.}R^2} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{Adj. } R^2} \tag{3.3.3.12}$$

where

$$j = \{1\text{LSR}_{noCV}, 1\text{LSR}_{CV}, 2\text{LSR}_{noCV}, 2\text{LSR}_{CV}\}. \tag{3.3.3.13}$$

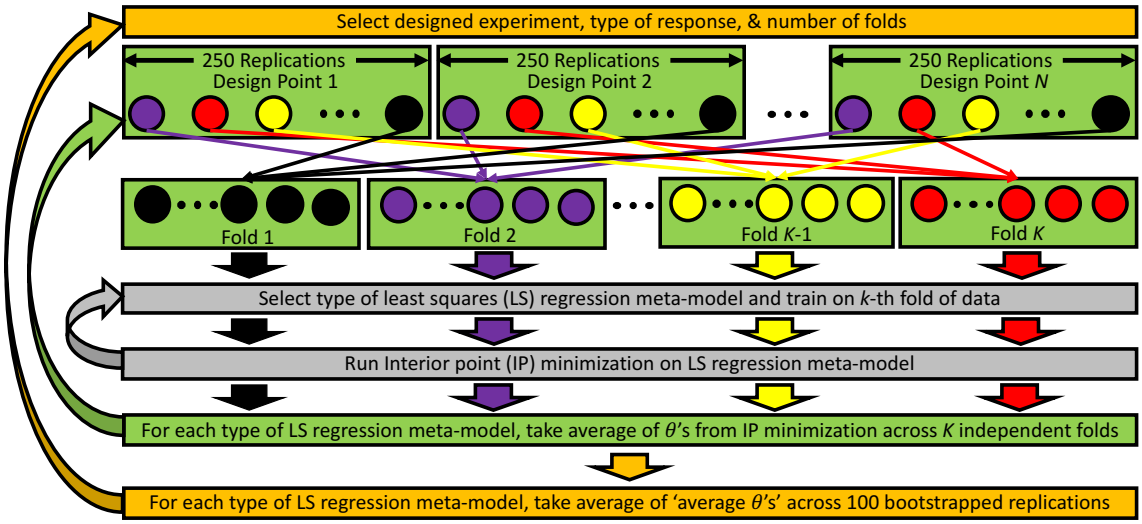


Figure 18. Abstract Flowchart for Multiple Population Regression Meta-experiment

The next section of this chapter will provide the results of a comparative analysis of the relative performance of the second order least squares regression model that includes CV interactions with the other three least squares regression metamodel formulations. However, because of the sheer size of the total meta-experiment study

(30 combinations of five designed experiments and 6 responses), a small subsample of the entire meta-experiment was selected to highlight the results of the overall experiment. The entire exposition covering all the meta-experiments can be found in Section A.5 of Appendix A.

3.4 Results

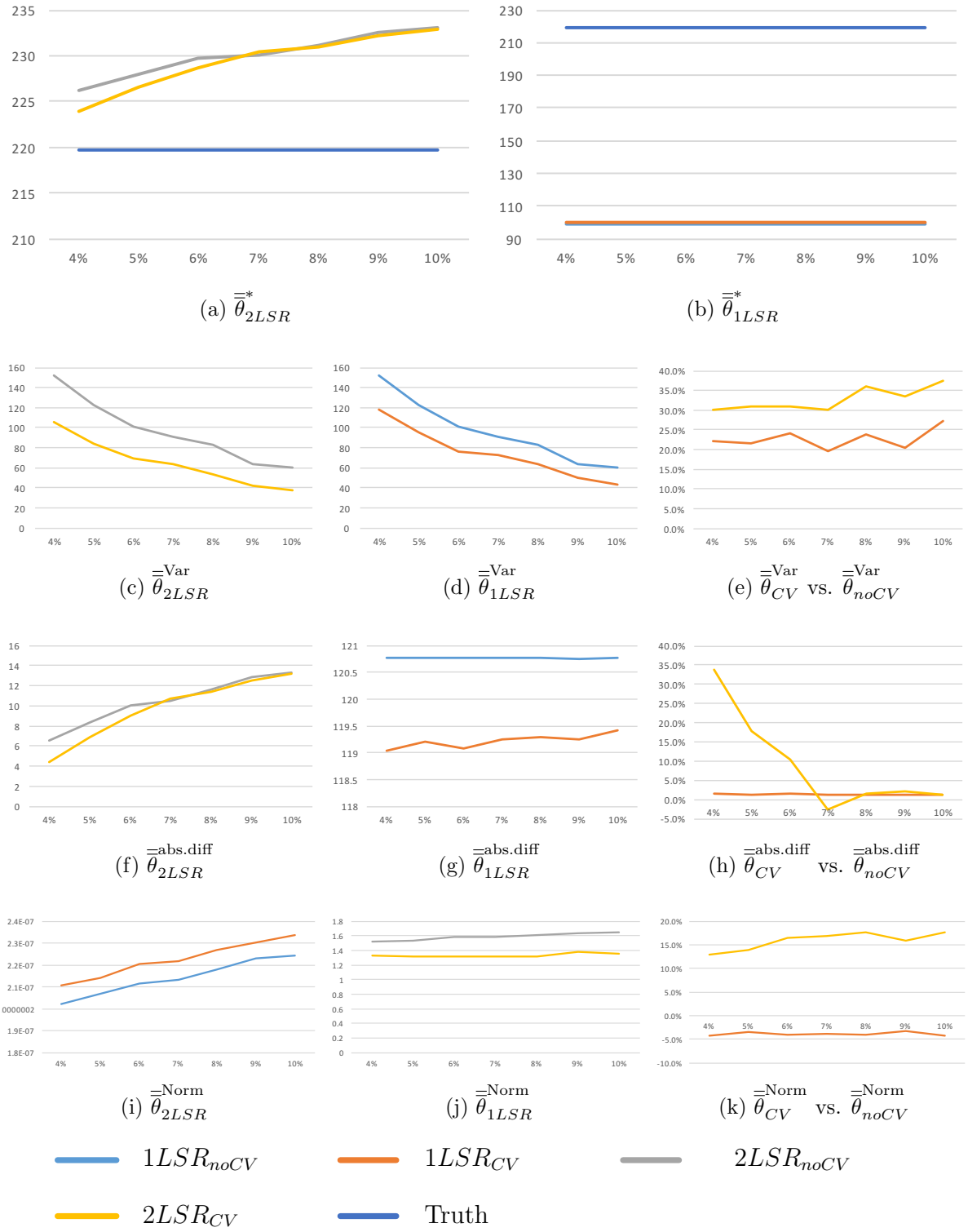
The objectives for this study is to first evaluate the utility of the second order least squares regression model that includes CV interactions in terms of how well it represents the true simulation responses. We contrast multiple statistics obtained from the second order least squares regression model with CV interactions with the statistics gathered from three other smaller least squares regression meta-models to evaluate its performance. The second objective to this study is to consider the information that is revealed by the introduction of the new interaction terms to strengthen the understanding of the system dynamics that are contributing to the simulation output of interest. The significance of the components are identified by evaluating the magnitude of the absolute value of the estimating coefficient and its associated statistical p value.

3.4.1 Second order Least Squares Regression Meta-model with CV Interaction Performance.

Subsection 3.3, describes the methodology for the meta-experiment that was conducted to evaluate the performance of the unbiased second order formulation of the least squares regression meta-model with control variate interactions. In this section, the results are summarized for the full factorial designed experiment as the type of output or nonlinearly transformed composition of the outputs and the size of the K

folds is changed or incremented, respectively. The performance of the four types of regression meta-models are contrasted as these components are interchanged.

Figures 19 - 22 contrast the performance of four types of least squares regression metamodels, $1LSR_{noCV}$, $1LSR_{CV}$, $2LSR_{noCV}$, and $2LSR_{CV}$; across a set of metrics for two responses, the average TV sojourn time, Y, and the second nonlinearly transformed cost function, CF_2. Subfigures (a) & (b) of Figures 19 and 20, provide the average minimum response, after an interior point optimization routine (see Section 1.2.5, across 100 replications of the number of K folds respective to the percentage of data that is in each fold (4% to 10%) and to the type of regression meta-model that is utilized. Subfigures (c) & (d), provides the average variance of the minimum response observed in Subfigures (a) & (b), respectively. Subfigure (e), gives the percent of increase or decrease in the observed variance from Subfigures (a) & (b), when CV models are compared with the associated no secondCV model. For example, the second order CV model is contrasted with the second order no CV model and the first order CV model is contrasted with the first order no CV model. Subfigures (f) & (g), provide the average absolute value of the difference between the minimum meta-model response, $\hat{Y}^*$, and the true minimum response, Y^*, that is derived from the Jackson formula analysis. Subfigure (h), gives the percent of increase or decrease in the absolute difference from Subfigures (f) & (g), when CV models are compared with the associated no CV model. Subfigures (i) and (j), provides the average $L2$ norm between the meta-model coordinates for $\hat{Y}^*$ and the Jackson formula coordinates for Y^*. Subfigure (k), gives the percent of increase or decrease in the observed L2 norm from Subfigures (i) & (j), when CV models are compared with the associated no CV model.



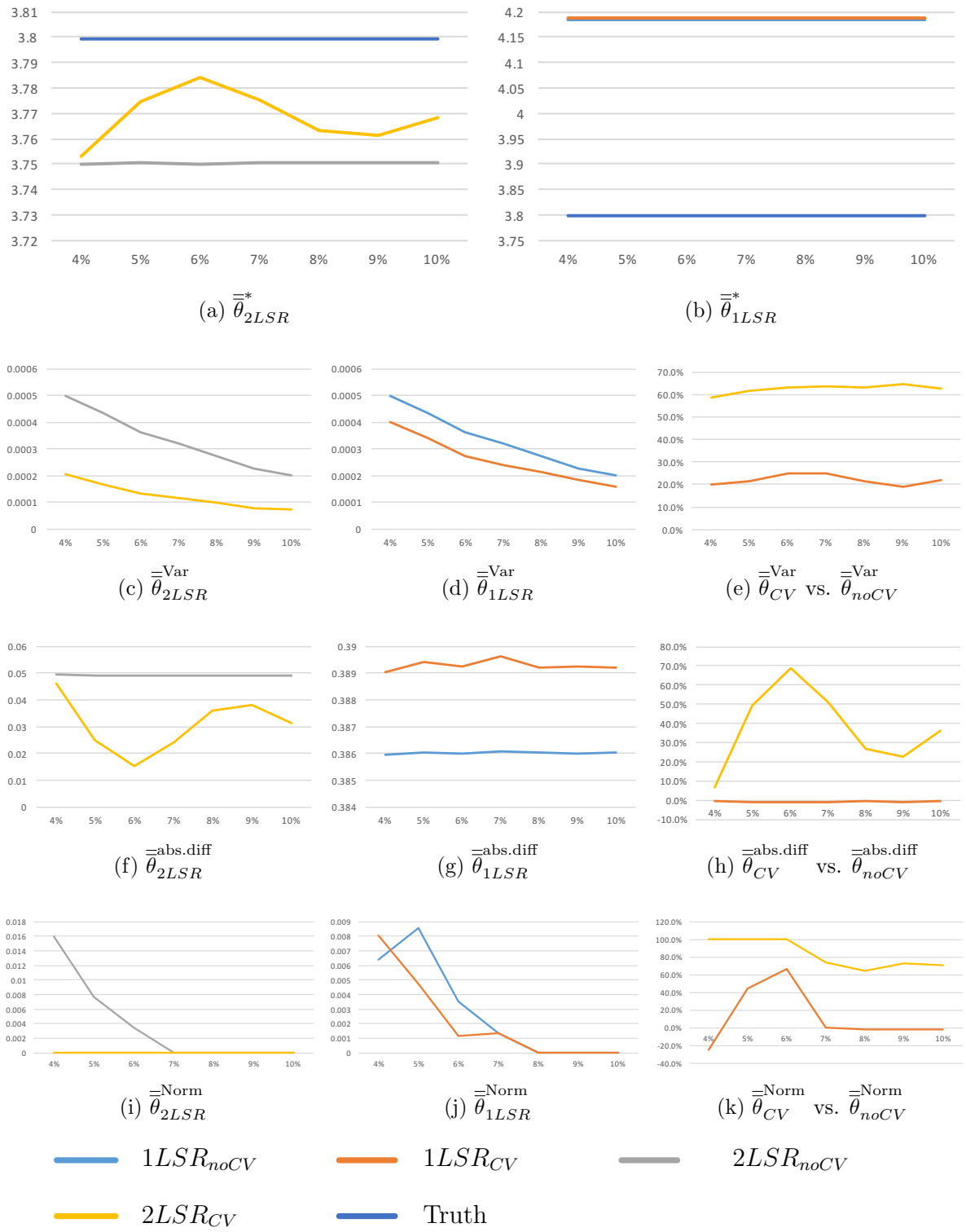
(a) $\overline{\overline{\theta}}^{*}_{2LSR}$ (b) $\overline{\overline{\theta}}^{*}_{1LSR}$

(c) $\overline{\overline{\theta}}^{\text{Var}}_{2LSR}$ (d) $\overline{\overline{\theta}}^{\text{Var}}_{1LSR}$ (e) $\overline{\overline{\theta}}^{\text{Var}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{Var}}_{noCV}$

(f) $\overline{\overline{\theta}}^{\text{abs.diff}}_{2LSR}$ (g) $\overline{\overline{\theta}}^{\text{abs.diff}}_{1LSR}$ (h) $\overline{\overline{\theta}}^{\text{abs.diff}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{abs.diff}}_{noCV}$

(i) $\overline{\overline{\theta}}^{\text{Norm}}_{2LSR}$ (j) $\overline{\overline{\theta}}^{\text{Norm}}_{1LSR}$ (k) $\overline{\overline{\theta}}^{\text{Norm}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{Norm}}_{noCV}$

Figure 19. Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time



(a) $\overline{\overline{\theta}}^{*}_{2LSR}$ (b) $\overline{\overline{\theta}}^{*}_{1LSR}$

(c) $\overline{\overline{\theta}}^{\text{Var}}_{2LSR}$ (d) $\overline{\overline{\theta}}^{\text{Var}}_{1LSR}$ (e) $\overline{\overline{\theta}}^{\text{Var}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{Var}}_{noCV}$

(f) $\overline{\overline{\theta}}^{\text{abs.diff}}_{2LSR}$ (g) $\overline{\overline{\theta}}^{\text{abs.diff}}_{1LSR}$ (h) $\overline{\overline{\theta}}^{\text{abs.diff}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{abs.diff}}_{noCV}$

(i) $\overline{\overline{\theta}}^{\text{Norm}}_{2LSR}$ (j) $\overline{\overline{\theta}}^{\text{Norm}}_{1LSR}$ (k) $\overline{\overline{\theta}}^{\text{Norm}}_{CV}$ vs. $\overline{\overline{\theta}}^{\text{Norm}}_{noCV}$

Figure 20. Design: FF - Full Factorial, Response: CF_2 - Cost Function 2

Subfigures (a), (b), and (c), of Figures 21 and 22 illustrate the average mean squared error (MSE), coefficient of determination (R^2), and adjusted R^2, respectively, across 100 replications of the number of K folds and to the percentage of data that is in each fold (4% to 10%) and type of regression meta-model.

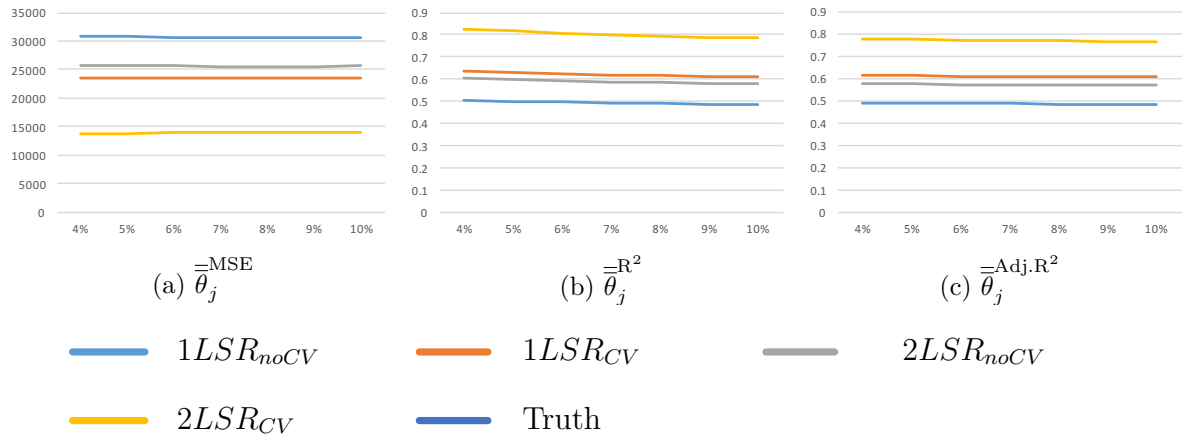


Figure 21. Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time

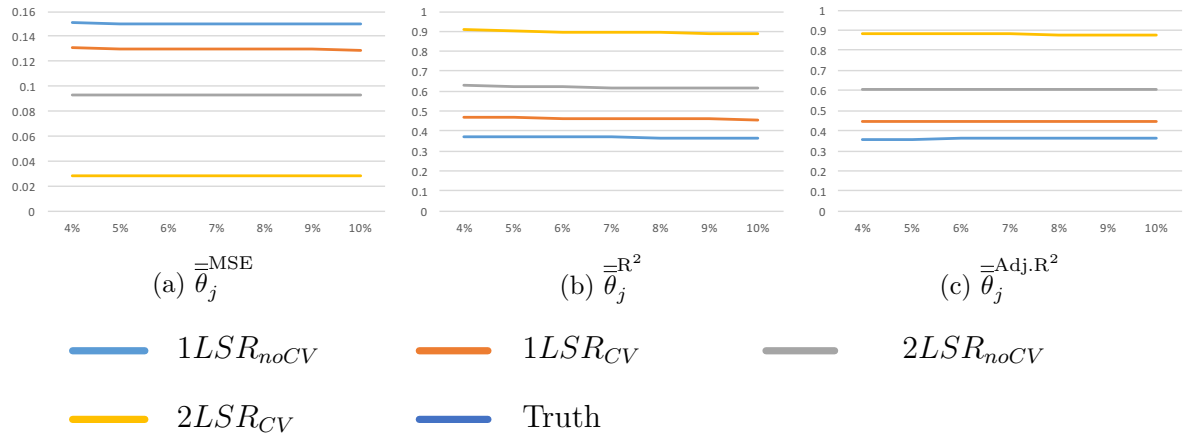
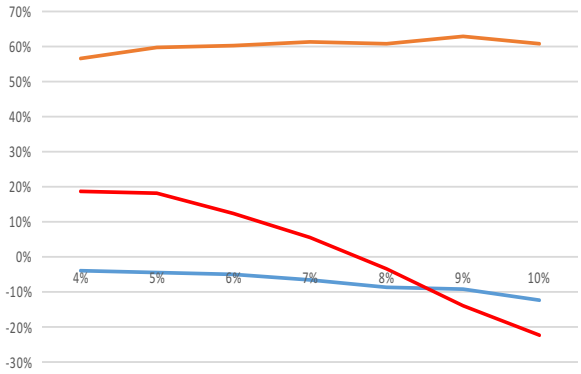
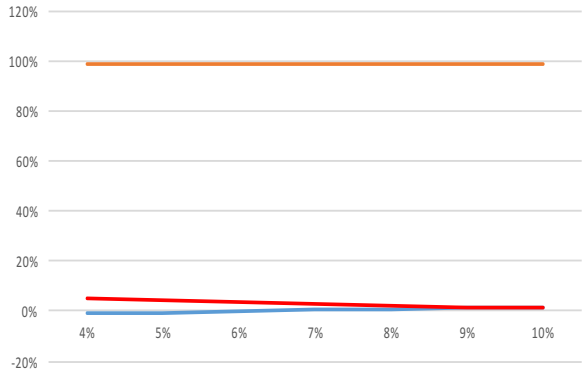


Figure 22. Design: FF - Full Factorial, Response: CF_2 - Cost Function 2

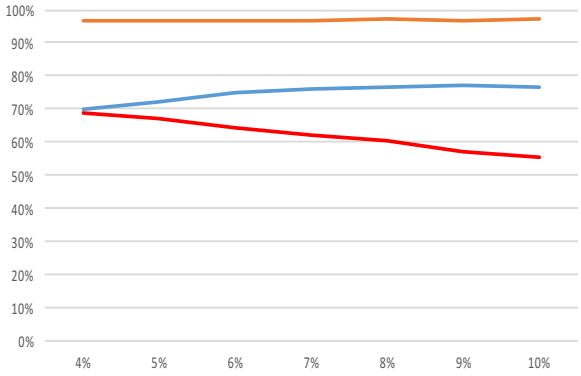
To compensate for the sheer size of the scope in this research, only the percent of increase or decrease in the observed variance, $\bar{\bar{\theta}}_{2LSR}^{\text{Norm}}$, L2 Norm, $\bar{\bar{\theta}}_{2LSR}^{\text{Var}}$, and average absolute difference, $\bar{\bar{\theta}}_{2LSR}^{\text{Abs.Diff}}$, for all six responses of the full factorial design is provided.



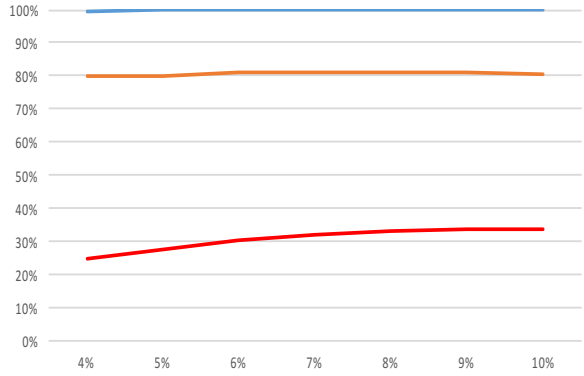
(a) Y: Average TV Sojourn Time



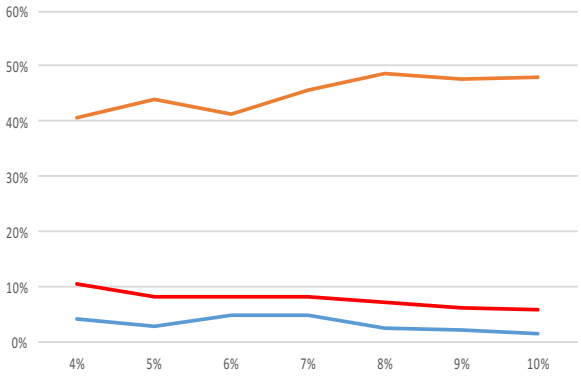
(b) U_A: Adjustor Queue Utilization



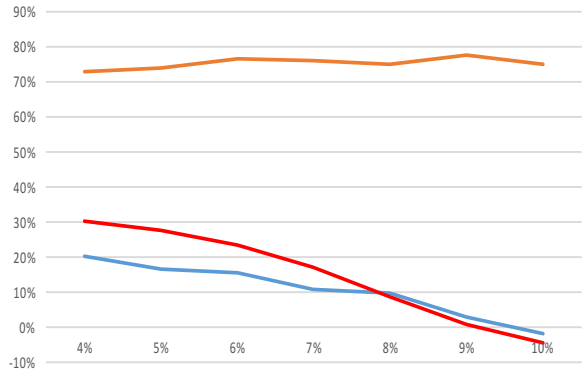
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



(e) CF_3: Cost Function 3



(f) CF_4: Cost Function 4

$\overline{=}$Norm $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Norm $\hat{\theta}_{2LSR_{noCV}}$ $\overline{=}$Var $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Var $\hat{\theta}_{2LSR_{noCV}}$ $\overline{=}$Abs.Diff $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Abs.Diff $\hat{\theta}_{2LSR_{noCV}}$

Figure 23. Design: FF - Full Factorial

By contrasting the meta-models, $2LSR_{noCV}$ with $2LSR_{CV}$, across Figures 19 - 23 a clear pattern emerges that suggests the introduction of the second order interaction terms to the existing second order control variate model proved to be robustly superior across the types of responses or the amount of data used to fit the the least squares regression model. See Section A.5 of Appendix A for comprehensive sets of tables and figures for all other meta-experiments conducted in this study.

3.4.2 System Dynamics.

We saw in Chapter II the unveiling of important signals that manifest from the second order interactions under the context of a single population simulation. The information from those signals provided more clarity to distinguishing the components of the simulation that were contributing a great deal of variance into the system, as the design point was changed to induce increasing amounts of variance. In a similar fashion, this section now extends the approach to the context under the context of multiple population simulation. Its noted that the perspective of the information is shifted from determining what is important in a simulation at a specific design point, to now determining what is important across an entire designed experiment through a range of design points.

By examining the absolute value of the coefficients that are statistically significant, $p < .05$, we can make inferences as to how the simulation components and their relationships with each other are contributing to the variance of the system responses of interest. In this experiment, the full factorial designed experiment, FF, with the corresponding non-linearly transformed response, CF_2, from equation (3.3.2.9) and Figure 17b, was selected to be featured here. As a direct result of the orthogonal property of the FF designed experiment, the independence of the CVs in W and R and the nature of their asymptotic standardization (see Table 13), we can evaluate

the magnitude of the absolute value of the coefficient in terms of how much variance the component contributes to the simulation. More precisely, the coefficient will tell us the effect that a one standard deviation change in a variable has on the mean response.

Table 13. Covariance of the FF, Full Factorial Designed Input Variates, X, and the Standardized "Work" and "Routing" Control Variates, W and R

COV{FF}	X_{λ}	$X_{\mu(I)}$	$X_{\mu(A)}$	$X_{p(FI)}$	W_{λ}	$W_{\mu(I)}$	$W_{\mu(A)}$	$R_{p(FI)}$
X_{λ}	1.00	0	0	0	0.02	0.02	-0.02	0.00
$X_{\mu(I)}$	0	1.00	0	0	-0.01	0.00	-0.01	0.01
$X_{\mu(A)}$	0	0	1.00	0	-0.01	-0.01	-0.03	0.01
$X_{p(FI)}$	0	0	0	1.00	0.01	-0.00	0.03	-0.03
W_{λ}	0.02	-0.01	-0.01	0.01	0.97	-0.00	0.01	-0.04
$W_{\mu(I)}$	0.02	0.00	-0.01	-0.00	-0.00	0.99	0.02	0.00
$W_{\mu(A)}$	-0.02	-0.01	-0.03	0.03	0.01	0.02	1.00	-0.01
$R_{p(FI)}$	0.00	0.01	0.01	-0.03	-0.04	0.00	-0.01	1.01

Referring to Figure 18, the third step of the methodology of the prior section, was taken to capture, $\{\hat{\theta}^{*}, \theta^{\text{abs.diff}}, \theta^{\text{Norm}}, \theta^{\text{MSE}}, \theta^{\text{R}^2}, \theta^{\text{Adj.}R^2}\}$. For the purposes of this specific experiment, an $N \times 1$ vector of statistics for the estimating coefficients, $\hat{\theta}^{\text{coef.}}$, and a $N \times 1$ vector of the associated p values, $\hat{\theta}^{\text{p}}$ for each of the four least squares regression models, $\{1LSR_{noCV},\ 1LSR_{CV},\ 2LSR_{noCV},\ 2LSR_{CV}\}$, where N is the number of estimating coefficients respective to the four regression models, was also recorded.

The average absolute value of the estimating coefficients that are statistically significant ($p < .05$) for all four least squares regression meta-models are summarized in Table 14 below. By contrasting horizontally, the viewpoint that is provided by including unbiased second order interactions of the standardized "work" and "routing" control variates with themselves and with the designed input variates to the other

meta-models that excludes the additional regressor terms, reveals a great deal of new information that was excluded in the smaller three meta-models.

Table 14. Regression Coefficients from All Meta-models for FF & CF_2, where $p > 0$

Regressor	$1LR_{noCV}$	$1LR_{CV}$	$2LR_{noCV}$	$2LR_{CV}$
0: Intercept	4.71	4.71	4.71	4.71
1: $\boldsymbol{X}_{\lambda}$	0.06	0.06	0.06	0.06
2: $\boldsymbol{X}_{\mu(I)}$	0.09	0.09	0.09	0.09
3: $\boldsymbol{X}_{\mu(A)}$	0.23	0.23	0.23	0.23
4: $\boldsymbol{X}_{\mathrm{p(FI)}}$	0.14	0.14	0.14	0.14
5: $\boldsymbol{W}_{\lambda}$				0.03
6: $\boldsymbol{W}_{\mu(I)}$				0.04
7: $\boldsymbol{W}_{\mu(A)}$		0.10		0.10
8: $\boldsymbol{R}_{\mathrm{p(FI)}}$		0.09		0.09
1 x 3			0.09	0.09
1 x 4			0.07*	0.06*
1 x 5				0.03
1 x 8				0.04
2 x 3			0.08	0.08
2 x 4				0.03
2 x 7				0.03
3 x 4			0.19*	0.20*
3 x 5				0.05
3 x 6				0.03
3 x 7				0.11
3 x 8				0.11
4 x 5				0.04
4 x 7				0.05
4 x 8				0.06
7 x 8				0.03

It is noted from Table 14, when the columns of coefficients are compared horizontally across the types of meta-models, $\{1LSR_{noCV}, 1LSR_{CV}, 2LSR_{noCV}\}$, the coefficients are largely unaffected (two negligible exceptions are marked by asterisk) by the inclusion of the additional CV interaction terms that is associated with the second order control variate meta-model, $2LSR_{CV}$. This is an important observation as the variance explained by the additional terms are, by and large, explaining far more of the unexplained variance than the other meta-models were capable of handling. This also is meaningful because it supports the covariance matrix from Table 13, in showing a lack of dependence between the terms.

It is also noted, the magnitude of the coefficients for the second order interactions between the designed input variable for the mean adjustor service time, $X_{\mu(A)}$, and standardized "work" variate for the adjustor service time, $W_{\mu(A)}$, and also with the standardized "routing" variate for the probability of failure, $R_{p(FI)}$, are greater in value then all but two of the first order coefficients, namely, $X_{\mu(A)}$ and $X_{p(FI)}$. This is also an important observation, in that it supports the intuition that in a system that is overloaded due to a nearly unstable adjustor queue ($0.85 < \rho < 1$) as is the case in the selected design region of the simulation, even a slight increase in workload will have a significant impact on the station that backs up the entire system, causing significant delay. The statistical significance of the second order interaction term between the standardized "work" variate for the adjustor service time, $W_{\mu(A)}$, with the standardized "routing" variate for the probability of failure, $R_{p(FI)}$, further supports this finding.

3.5 Conclusions

In this chapter, new statistical framework has been developed to extend the conventional second order least squares regression metamodel with first order linear con-

trol variates to now incorporate second order interaction terms between the control variates, and the interaction terms between the control variates and the designed input variables. By excluding the pure quadratic terms, this new formulation retains the unbiased feature found in the original formulation.

A study that experimented across 210 meta-experiments on five designed experiments and six types of outputs showed this new formulation robustly outperforms the existing second order formulation across a set of standard metrics. Further, a second objective demonstrated the enhanced insight into the governing system dynamics of the simulation. This information was gleaned from simply analyzing the magnitude of the absolute value of the statistically significant coefficients for the second order interactions within the CVs and the interactions of CVs and the designed input variables.

IV. Improved Neural Network Metamodels for Simulations Using Linear Control Variates

4.1 Introduction

In this section, a novel approach for using radial basis neural networks (RBNNs) as a meta-model for simulation is presented. The method introduces specialized control variates, called standardized "work" and "routing" variates (CVs) (see Sections 1.2.1.8 and 1.2.1.9), as additional inputs to the network. A heuristic is developed to quickly approximate the optimal number of neurons and the degree of spread for the RBNN architecture. In training, the "work" and "routing" CVs are introduced as additional dimensions to the pre-existing designed input variables that are used in the traditional process of training an RBNN. An extensive case study is conducted on a simulation (see Sections 1.2.6) that contrasts the predictive performance of an RBNN that *includes* CVs with an RBNN that *excludes* CVs, for which the latter approach is common in literature. The findings in this study suggests this new approach is orders of magnitude superior than the conventional RBNN method.

4.2 Extended Radial Basis Neural Network with CVs

The radial basis neural network (RB_{noCV}) formulation that excludes control variates (see Section 1.2.2.1) is extended here to include asymptotically standardized "work" and "routing" CVs. Let H be a $m \times 1$ column array of exponential basis function neurons, given as

$$H_i(X) = \exp\left(\frac{-D_i^2}{2\sigma^2}\right), \tag{4.2.0.1}$$

where equation 1.2.2.4 from Section 1.2.2.1 is now extended to,

$$D_i^2 = \left(\begin{bmatrix} X \\ Z \end{bmatrix} - \mu_i \right)^T \cdot \left(\begin{bmatrix} X \\ Z \end{bmatrix} - \mu_i \right), \tag{4.2.0.2}$$

and Z is a $(p+q) \times 1$ column vector made up of a $p \times 1$ column vector of standardized "work" CVs, C, that is vertically concatenated with a $q \times 1$ column vector of standardized "routing" CVs, R, X is a $n \times 1$ column vector of designed input variables, σ and μ are a $m \times 1$ column vectors of spreads and centers, respectively, that are associated with H. Then we can define the univariate output, Y, for the RBNNCV as

$$Y = \sum_i H_i \cdot W_i + B, \tag{4.2.0.3}$$

where W is a $m \times 1$ column vector of weights associated with H, and B is the bias that is learned from the iterative updates of W. Figure 24, an extension of Figure 6 from Section 1.2.2.1, illustrates the extended RBNNCV with an n dimensional vector of designed input variables, X, a $(p+q)$ dimensional vector of standardized "work" and "routing" control variates, Z, an array of m exponential basis functions, H, a vector of m weights for each of the basis functions, W, and the single node Σ for the output, Y, with bias, B.

See equations (1.2.2.6) and (1.2.2.7) of Section 1.2.2.1 for RBNN algorithms on learning the values of the elements in the weight vector, W, and the network bias, B, respectively.

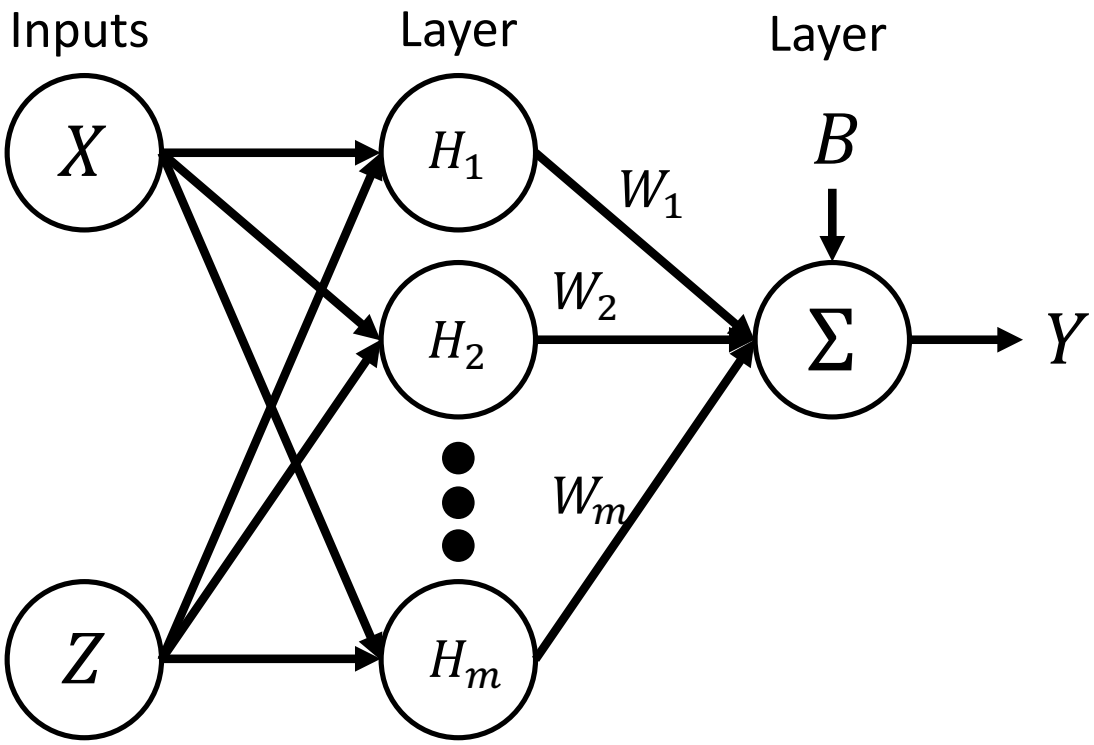


Figure 24. Example Extended Radial Basis Neural Network with CVs (RB_{CV})

4.3 Case Study

This section provides analysis from the study that demonstrated that the radial basis neural network that includes asymptotically standardized "work" and "routing" CVs, can significantly outperform, the radial basis neural network that excludes asymptotically standardized "work" and "routing" CVs. First, a heuristic is developed to identify approximate optimal architectures for the various radial basis neural networks that are constructed. After identifying these conditions, the two neural networks with different input configurations are tested against five designed experiments and six responses from a computer simulation characterized by a high degree of system variance. Two of the six tested responses, which are the average sojourn time, Y, and the adjustor queue utilization, U_A, are drawn directly from the simu-

lation. The remaining four responses, CF_1, CF_2, CF_3, CF_4, take the two outputs from the simulation and pass them through an additional layer of non-linearity. That is, they are passed into four different types of non-linear transformations where the single transformed response from each transformation is subsequently trained on by the radial basis neural networks.

In practice, developing a meta-model isn't the end state for the simulation practitioners. A common goal is to utilize the meta-model as a means of identifying an estimate for the optimal simulation configuration, which will return a minimum (or maximum) expected value for a response of interest from the underlying simulation. This consideration, takes the case study to one final step, which is to minimize the expected value of the trained radial basis neural network. However, before the minimization can occur, a method used by Bellucci [18] under the context of robust parameter design, approximates the expected value of a trained radial basis neural network by taking a Taylor series approximation of the network. This step produces a second order function that is by definition a convex function. As a result, the second order approximation can be minimized by any type of convex non-linear minimization techniques, namely, interior-point minimization that is used in this research.

The sequence of steps in this case study are encapsulated by,

$$\begin{aligned} \text{Minimize} \quad & \hat{\mathrm{E}}_{Taylor}\left(RBNN(S) \mid S \in \left\{\bar{Y}, \bar{U}_A, CF_i\left\{[\bar{Y} \quad \bar{U}_A]\right\}\right\}, \text{ for } i = 1, 2, \ldots, 4\right) \\ \text{s.t.} \quad & [X^*_{\lambda}, X^*_{\mu(I)}, X^*_{\mu(A)}, X^*_{p(FI)}] \in \text{Design Space} \end{aligned}$$

where, CF_i, is the cost function that represents one of the four non-linear transformations of the two responses, $\bar{Y}$ and $\bar{U}_A$. $RBNN$, represents the radial basis neural network training on one of the two simulation responses, $\bar{Y}$ and $\bar{U}_A$, or one of the four non-linear transformations, CF_i. The estimated expected value that is derived

from the Taylor series approximation of the trained radial basis neural network is annotated by, $\hat{\mathrm{E}}_{Taylor}$. The minimization that is invoked via interior point method is bounded by the predicted optimal simulation configuration, $[X^*_{\lambda}, X^*_{\mu(I)}, X^*_{\mu(A)}, X^*_{p(FI)}]$, which must reside in the selected design space. The results that are derived from these steps are then compared with the true minimum mean long run responses that are evaluated by Jackson open network formulas.

4.3.1 Simulation Analysis.

In a manner similar to the case study of Chapter III on second order CVs for a least squares regression meta-model, a long run truth model was computed for a simulation (see Section 1.2.6) that is characterized by the properties of an open Jackson network. The use of Jackson open network formulas (see Section 1.2.5) acted as the baseline for any metrics that are formed in this study. The Jackson formulas were used to first identify a feasible design region with the condition that the inspector and adjustor queues were stabilized and also exhibiting an exceptional high variance, i.e. $0.85 <= \lambda/\mu = \rho < 1$. See Tables 9 and 10 in Section 3.3, that summarize the range of the input settings and simulation outputs. Also see Table 17 in Appendix A.1 for more specific information on Jackson formula evaluations for each design setting.

After the design space was identified, five types of designed experiments (summarized in Table 11 of Section 3.3) were conducted on the simulation that is described in Section 1.2.6 of Chapter I. The five selected designs, $\{FF$,CCc,CCf,BB,$LH\}$ (see Table 11 of Section 4.3.1), are comprised of 107 total simulation design points. 34 of the 107 total design points are duplicated across multiple designed experiments, (see Table 17 of Appendix A.1). The remaining 73 non-overlapping, unique design points were simulated. Each design point is replicated 250 times, providing a total of 18,250 data points. The steady-state start time and simulation stop time for each

replication across all configurations is 250,000 and 500,000 time units, respectively. All information in the transient period (that was below 250,000 time units) was eliminated from the two simulation outputs of interest, $\bar{Y}$ and $\bar{U}_A$, the three standardized "work" CVs, W_λ, $W_{\mu(I)}$, $W_{\mu(A)}$, and the single standardized "routing" CV, $R_{p(FI)}$.

In an effort to validate the simulation, an exercise in Chapter II contrasted the long run mean sojourn time observed in the simulation at exceptionally long run lengths (up to 700 million time units) with the true long run mean sojourn time evaluated from Jackson formulas. Table 2 of Section 2.3.2 summarizes 10 simulation design points (that are different design points from the 73 design points that are used in Chapter III and this chapter) that increment the amount of system variance that is induced into the simulation at each subsequent design point. For example, the first design point of the experiment evaluated the simulation at $X_{p(FI)} = 0.05$ which resulted in a adjustor queue utilization of $\rho_{\mu(A)} = 0.105$ and very little system variance. The failure rate was incremented by 5% for the next five design points until $X_{p(FI)} = 0.30$ which resulted in an adjustor queue utilization of $\rho_{\mu(A)} = 0.857$, providing more system variance but relatively low in relation to the next 4 design points. The next three design points incremented the failure rate by 1% until $X_{p(FI)} = 0.33$ which resulted in an adjustor queue utilization of $\rho_{\mu(A)} = 0.985$, and the system variance is now relatively high in relation to the first five design points. One additional, tenth design point incremented the failure rate by 0.003% to $X_{p(FI)} = 0.333$ which resulted in the utilization of the adjustor queue to be $\rho_{\mu(A)} = 0.999$. At this final setting, the system variance that was observed in the simulation was extremely high. Also see Tables 17 through 20 in Appendix A.A.1 that provide specific information for each of the 73 total simulation design points used throughout the research in this document. The information includes, the type of design(s) a specific setting is associated with, the coded input setting, and the observed long run responses for average sojourn time

and the average adjustor station utilization rate across the 250 replications of each configuration from the simulation, and it's associated *true* long run values when the setting is evaluated through Jackson open network formulas.

See Figures 14a and 14b in Section 3.3 for an illustration of the projections of the four dimensional input space, to the two dimensional output space, Y and U_A, through the Jackson formulas (annotated as red circles), or through the simulation (annotated as blue circles), $\bar{Y}$ and $\bar{U}_A$. The mean of the observed responses, $\overline{\overline{Y}}$ and $\overline{\overline{U}}_A$, across the 250 replications of its respective setting is also illustrated as black squares. The feasible true mean surface region, associated with the Jackson formulas, is represented by the green dots.

Following the methodology detailed in Chapter III, the true long run sojourn time and true long run adjustor station utilization from the Jackson formulas, Y and U_A, or the observed long run sojourn time and observed adjustor station utilization from the simulation, $\bar{Y}$ and $\bar{U}_A$, are nonlinearly transformed through four different nonlinear "cost" functions, CF_1 (see equation (3.3.2.8) and Figure 17a), CF_2 (see equation (3.3.2.9) and Figure 17b), CF_3 (see equation (3.3.2.10) and Figure 17c), and CF_4 (see equation (3.3.2.11) and Figure 17d).

A meta-experiment was conducted on the data that was accumulated from the simulation across all the design settings associated with a particular designed experiment. For each designed experiment, various characteristics of the simulation data were varied to inspect if any significant effects were evident. These varied components of the meta-experiment are:

- # of folds: $K = \{25, 20, 17, 14, 13, 11, 10\}$
- selected designed experiment, the
  - FF: 2^4 Full Factorial

  - CCf: Central Composite - Faced
  - CCc: Central Composite - Circumscribed
  - BB: Box-Behnken
  - LH: Latin Hypercube

- type of output or nonlinearly transformed composition of outputs, and the
  - $\bar{Y}$: Average TV Sojourn Time
  - $\bar{U}_A$: Adjustor Server Utilization
  - CF_1: Cost Function 1
  - CF_2: Cost Function 2
  - CF_3: Cost Function 3
  - CF_4: Cost Function 4

- type of radial basis neural network metamodel invoked.
  - RB_{noCV}: Radial Basis Neural Network *excluding* CVs
  - RB_{CV}: Radial Basis Neural Network *including* CVs

The first step of the meta-experiment was to select one of the five designed experiments, $\{FF, CCf, CCc, BB, LH\}$, and one of the six responses, $\{Y, U_A, CF_1, CF_2, CF_3, CF_4\}$, that will be modeled in the selected meta-experiment.

4.3.2 Approximate Optimal Radial Basis Neural Network Spread & Neuron Architecture Heuristic.

Figure 25 gives an illustration of the k-fold cross validation heuristic that was developed to determine an approximate optimal architecture for the RBNNCV used in the next phase of this case study.

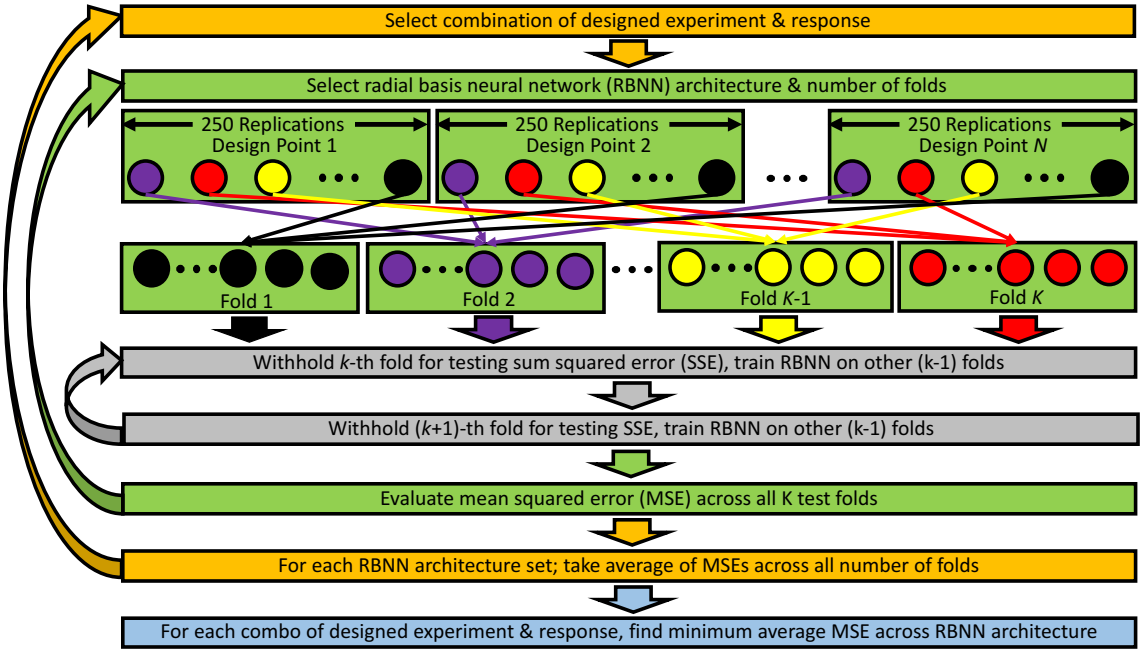


Figure 25. Abstract Flowchart for Approximate Optimal RB_CV Spread & Neuron Architecture Heuristic

The first step, illustrated in the top orange bar of Figure 25, is the selection of one of the five designed experiments, $\{FF, CCf, CCc, BB, LH\}$, and one of the six responses, $\{Y, U_A, CF_1, CF_2, CF_3, CF_4\}$.

The second step, illustrated in the top green bar of Figure 25, is the selection of one of the cross-validation sets of $K = \{25, 20, 17, 14, 13, 11, 10\}$ folds, and a selected radial basis neural network architecture for spreads, $S = \{1, 1.5, 2, 2.5 \ldots, 10\}$, and number of neurons, $N = \{1, 2, \ldots, 75\}$. These folds also represent the equal distribution of a selected number of data points from each design point (i.e. purple colored circles represents all selected observations from a given design point) into one of the K folds (i.e. all purple dots accumulated into same fold). For example, if the selected number of cross-validation folds is 13, then $\lfloor 250/13 \rfloor = 19$ replications (when rounded down) from each of the design points of a designed experiment is placed into a single fold. In this example, if the selected design was the 2^4 full factorial, FF, with 16 design points (see Tables 17 - 20 in Appendix A.1), then each fold would receive

$16 \cdot 19 = 304$ total data points. Table 12 of Section 4.3.1, gives a summary of the results of this method across all the sets of cross-validation folds. The reasoning for the selection of this range of folds is similar to Chapter III, in that often times, simulation practitioners have limited resources, and the intent of this study is to provide results for a scenario that would be reasonable in its application.

The third step, after the selection of the type of design, response, architecture, and number of folds, is the even distribution of data which is illustrated in Figure 25, directly below the top green bar. For example, in this step a purple dot from each of the design points are all selected from the same relative position in each design point and placed into a single fold that receives only purple dots. This step iteratively cycles through all K folds, until all data from each design point of the simulation is distributed out to K independent cross-validation folds.

The fourth step, illustrated in the two grey bars of Figure 25, encompasses three parts:

- the selection of one of the two types of radial basis neural network meta-models, $\{RB_{noCV}, RB_{CV}\}$,
- training selected radial basis neural network meta-model on the selected $(k-1)$ folds, and
- testing selected radial basis neural network meta-model on the k-th withheld fold where the controls are set to zero, $Z = 0$.

A particularly important note is made here with regards to the third part of this fourth step. This heuristic is an exhaustive procedure that requires a significant investment of computational resources, before the case study could be moved to the second phase. The objective was to derive the optimal architectures for spread and number of nodes, for both RB_{noCV} and RB_{CV}, for all combinations of designed

experiments and type of response. This requires the expected value of the network which can be estimated through two avenues. The first approach, tests a trained radial basis neural network when the controls are all set to zero, $Z = 0$. This approach is the fastest because it avoids taking the second order Taylor series approximation of the network. However, this method is likely biased by assuming their are no dependencies in the network on the control variates. The second approach, tests the second order Taylor series approximation of the radial basis neural network, which requires the calculation of the Hessian for the network. This second approach, is the slowest of the two methods, but would produce a more accurate estimate of the expected value of the radial basis neural network.

So in an effort to reduce the burden of calculating the Hessian thousands of times in this heuristic if the second order Taylor series approximation was selected, it was assumed that the optimal architecture that results from the first option would be nearly equivalent to the optimal architecture that results from the second. A small ancillary experiment was conducted to partially verify this assumption, in an effort, to show that the optimal architectures for S and N, would be reasonably similar at each of their respective minimum MSE values. The full factorial designed experiment, FF, and second cost function, CF_2, was selected as the components for this experiment. The minimum MSE was evaluated for both types of RB_{CV} formulations across all values of $S = \{1, \ldots, 10\}$ and $N = \{1, 2, \ldots, 75\}$ and illustrated in Figure 26 below.

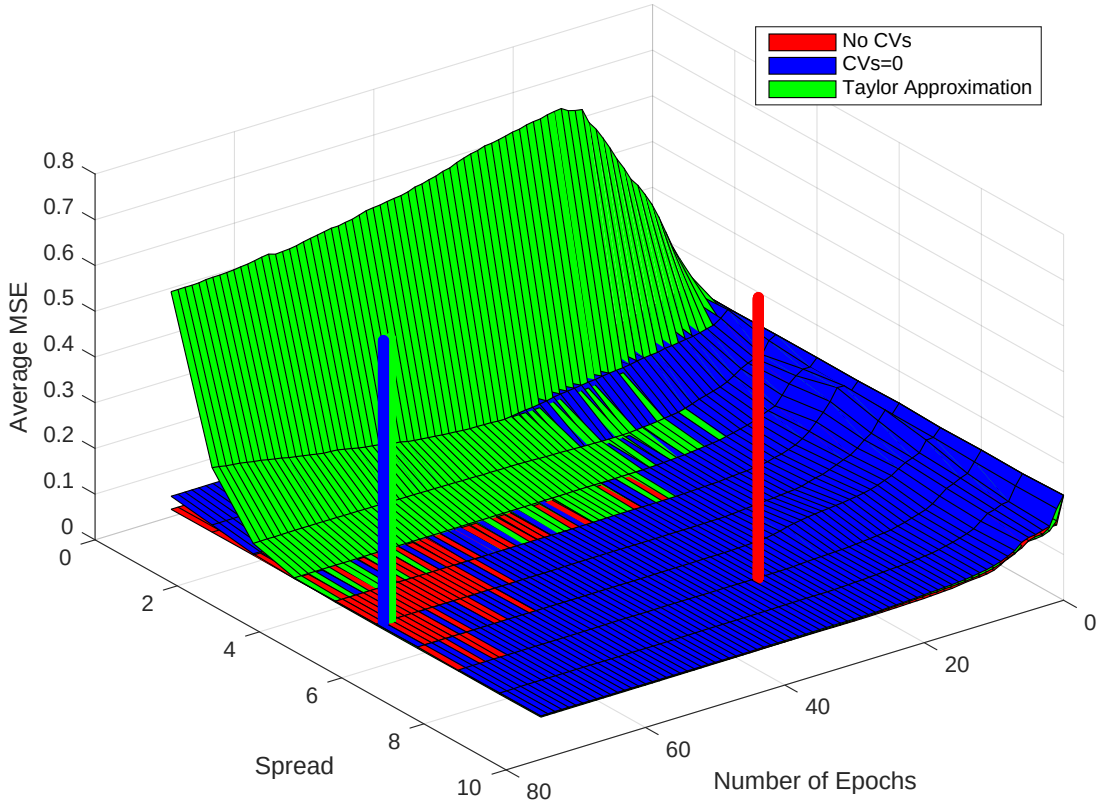


Figure 26. Results for RB_{CV} Assumptions Experiment, Design: FF - Full Factorial, Response: CF_2 - second Cost Function

The three surfaces represent the MSE (z-axis) that is evaluated from the selected architecture S (x-axis) and N (y-axis) for the RB_{noCV} (Red) that excludes the CVs and the two RB_{CV} formulations ($Z = 0$ is blue or second order Taylor series approximation that is green) that include CVs. The three tall lines protruding from the surfaces pinpoint the minimum MSE that is observed for its respective type of radial basis neural network, where the green line is the minimum MSE for the Taylor series approximation of the RB_{CV}, blue line is the minimum MSE for the RB_{CV} evaluated at $Z = 0$, and the red line is the minimum MSE for the RB_{noCV}.

It is clear in Figure 26 that there was a significant difference in the optimal architectures of the RB_{noCV} ($S = 8$, $N = 32$) and the optimal architecture for the

two formulations of the RB_{CV}. However, between the two RB_{CV} formulations, the location of the minimum MSE for the Taylor series approximation of the RB_{CV} at ($S = 6$, $N = 74$), was very close to the location of the minimum MSE for RB_{CV} that is evaluated at $Z = 0$ ($S = 6$, $N = 73$). This provides support to the conclusion of the assumption made in the third part of the fourth step to avoid calculating the Hessian many times for this heuristic.

Returning to the fourth step, this heuristic is accomplished for both radial basis neural network meta-model formulations, $\{RB_{noCV}, RB_{CV}\}$, and through all K cross-validation folds, recording the sum of squared errors (SSE) statistic, $\theta^{SSE}_{d,r,S,N,j,K,k}$, where $d = \{FF, CCc, CCf, BB, LH\}$ is the type of design, $r = \{Y, U_A, CF_1, CF_2, \ldots CF_3, CF_4\}$ is the type of response, $S = \{1, 1.5, 2, 2.5 \ldots, 10\}$ is the spread architecture, $N = \{1, 2, \ldots, 75\}$ is the number of nodes architecture, $j = \{RB_{noCV}, RB_{CV}\}$ is the type of radial basis neural network, total number of $K = \{25, 20, 17, 14, 13, 11, 10\}$ folds, and $k = \{1, 2, \ldots, K\}$ is the withheld fold.

The fifth step that is illustrated in the bottom green bar of Figure 25, calculates the mean squared error (MSE) across the number of withheld folds, and the sixth step that is illustrated in the bottom orange bar, evaluates the average MSE across the sets of $K = \{25, 20, 17, 14, 13, 11, 10\}$ folds, where both are given as,

$$\bar{\theta}^{\text{MSE}}_{d,r,S,N,j} = \frac{1}{7} \cdot \sum_{i=1}^{7} \frac{1}{K} \cdot \sum_{k=1}^{K} \theta^{SSE}_{d,r,S,N,j,k}, \tag{4.3.2.1}$$

where $d = \{FF, CCc, CCf, BB, LH\}$ is the type of design, $r = \{Y, U_A, CF_1, \ldots CF_2, CF_3, CF_4\}$ is the type of response, $S = \{1, 2, \ldots, 10\}$ is the spread architecture, $N = \{1, 2, \ldots, 75\}$ is the number of nodes architecture, $j = \{RB_{noCV}, RB_{CV}\}$ is the type of radial basis neural network, $k = \{1, 2, \ldots, K\}$ is the withheld fold, and $K = \{25, 20, 17, 14, 13, 11, 10\}$ is the total number of folds.

Before moving to the sixth step, the fifth step concludes by iteratively cycling back to the second step, to select the next spread value, $S = \{1, 2, \ldots, 10\}$, the next number of nodes architecture, $N = \{1, 2, \ldots, 75\}$, and number $K = \{25, 20, 17, 14, 13, 11, 10\}$ folds, until all values of S, N, and K have been exhausted. Before moving to the final step, the sixth step concludes by iteratively cycling back to the first step, to select the next design, $d = \{FF, CCc, CCf, BB, LH\}$, and type of response, $r = \{Y, U_A, CF_1, CF_2, CF_3, CF_4\}$, until all values of d and r have been exhausted.

The sixth and final step in this heuristic, illustrated in the bottom blue bar of Figure 25, evaluates the minimum $\bar{\theta}^{\text{MSE}}_{d,r,S,N,j}$ for each type of design, $d = \{FF, CCc, \ldots CCf, BB, LH\}$, type of response, $r = \{Y, U_A, CF_1, CF_2, CF_3, CF_4\}$, and type of radial basis neural network, $j = \{RB_{noCV}, RB_{CV}\}$. The values of the spread, $S = \{1, 1.5, 2, 2.5 \ldots, 10\}$, and the number of neurons, $N = \{1, 2, \ldots, 75\}$, is recorded for each minimum MSE to be used as the settings for the second phase of this case study. The optimal architectures for S^* and N^*, respective to the type of designed experiment, type of responses, and type of radial basis neural network is summarized in Tables 15 and 16.

Table 15. Approximate Optimal Radial Basis Neural Network including CVs (RB_{CV}) Architecture

	S^* - Spread					N^* - Number of Nodes				
Response	FF	CCc	CCf	BB	LH	FF	CCc	CCf	BB	LH
Y	5.5	8.0	10.0	8.0	9.5	36	36	41	44	33
U_A	9.5	10.0	10.0	9.5	10.0	70	90	41	66	23
CF_1	10.0	10.0	10.0	10.0	9.5	42	64	46	64	52
CF_2	5.5	5.5	4.5	5.5	3.5	62	70	77	70	46
CF_3	7.5	8.0	7.5	8.0	8.5	45	47	46	47	27
CF_4	10.0	9.5	10.0	9.5	9.5	33	38	34	38	36

Table 16. Approximate Optimal Radial Basis Neural Network excluding CVs (RB_{noCV}) Architecture

	S^* - Spread					N^* - Number of Nodes				
Response	FF	CCc	CCf	BB	LH	FF	CCc	CCf	BB	LH
Y	4.0	5.5	4.5	5.5	6.0	28	25	30	33	24
U_A	5.5	6.0	6.5	6.0	5.5	42	39	28	31	20
CF_1	5.0	4.5	5.5	6.0	5.5	26	28	37	41	36
CF_2	4.0	4.5	4.5	5.5	3.5	44	37	47	32	45
CF_3	4.0	4.0	3.5	5.0	5.5	39	25	34	22	45
CF_4	6.0	6.0	5.5	6.0	5.5	31	27	31	24	32

4.3.3 Meta-Experimentation for Radial Basis Neural Network Meta-models.

In a similar process to the meta-experimentation from Section 3.3.3 of the previous chapter, a slightly modified meta-experiment was conducted on the same set of data as the prior chapter. The data was accumulated from the simulation across all the design points associated with a particular designed experiment. For each designed experiment, various characteristics of the simulation data were varied to ascertain if any significant affects were evident. These varied components of the modified meta-experiment are:

- # of folds: $K = \{25, 20, 17, 14, 13, 11, 10\}$
- selected designed experiment
  - FF: 2^4 Full Factorial
  - CCf: Central Composite - Faced
  - CCc: Central Composite - Circumscribed

  - BB: Box-Behnken
  - LH: Latin Hypercube

- type of output or nonlinearly transformed composition of outputs
  - Y: Average TV Sojourn Time
  - U_A: Adjustor Server Utilization
  - CF_1: Cost Function 1
  - CF_2: Cost Function 2
  - CF_3: Cost Function 3
  - CF_4: Cost Function 4

- type of radial basis neural network meta-model invoked
  - RB_{noCV}: radial basis neural network meta-model *excluding* CVs
  - RB_{CV}: radial basis neural network meta-model *including* CVs

Figure 27 illustrates the steps of the modified meta-experiment that was conducted as part of the study in this chapter. The first step, illustrated in the top orange bar of Figure 27, was the selection of one of the five designed experiments, $\{FF, CCf, CCc, BB, LH\}$, one of the six responses, $\{Y, U_A, CF_1, CF_2, CF_3, CF_4\}$, and one of the sets $K = \{25, 20, 17, 14, 13, 11, 10\}$ cross-validation folds. These folds also represent the number data points from each design point into one of the folds. See the description of meta-experimentation from previous chapter in Section 4.3.1 for a brief example of this step. Also see Table 12 of Section 4.3.1 for summary of this calculation for all scenarios. The reasoning from the heuristic procedure of the prior section, or the meta-experimentation of Chapter III applies to this procedure of evaluating the performance of the two types of radial basis neural networks, under a scenario that would be reasonable in application.

The second step of this procedure, after the selection of the type of design, response, and number of folds, is the even distribution of data which is illustrated in Figure 27, directly below the top orange bar. See the description of meta-experimentation from previous chapter in Section 4.3.1 for a brief example of this step. This step cycles through all K folds, until all data from each design point of the simulation is distributed out to K independent cross-validation folds.

The third step illustrated in the three grey bars of Figure 27 encompasses four steps:

- the selection of one of the two types of radial basis neural network meta-models, $\{RB_{noCV}, RB_{CV}\}$,
- training selected radial basis neural network meta-model on the nine cross-validation sets of data that were not withheld for testing,
- taking second order Taylor series approximation of the neural network meta-model, and
- invoking interior-point minimization routine on the second order approximation of the trained regression meta-model.

To address the the third part of this third step, taking the second order Taylor series approximation of a trained radial basis neural network is an approach used by Bellucci [18] under the context of robust parameter design for simulation. Bellucci's technique is used in this research, primarily because the interior-point minimization that is also used in this research is a non-linear optimization technique appropriate only for convex functions. The interior-point method would not perform well directly on a trained radial basis neural network, because the networks are non-convex. The second order Taylor series approximation of a trained radial basis neural network

are convex by definition, and would be an appropriate function for the interior-point minimization technique to be used on.

Bellucci reviews the multivariate delta method, which gives the expected value of a function for the random vector Z, as

$$\mathrm{E}\{Y\} = \mathrm{E}\{f(Z)\} \tag{4.3.3.1}$$

$$= \mathrm{E}[f(\mu_Z) + \nabla f(\mu_Z)' \cdot (Z - \mu_Z) + \frac{1}{2} \cdot (Z - \mu_Z)' \cdot \mathrm{H} \cdot (\mu_Z) \cdot (Z - \mu_Z)] \tag{4.3.3.2}$$

$$= f(\mu_Z) + \nabla f(\mu_Z)' \cdot \mathrm{E}[(Z - \mu_Z)] + \frac{1}{2} \cdot \mathrm{E}[(Z - \mu_Z)' \cdot \mathrm{H} \cdot (\mu_Z) \cdot (Z - \mu_Z)] \tag{4.3.3.3}$$

$$= f(\mu_Z) + \frac{1}{2} \cdot tr(\mathrm{H}(\mu_Z)\Sigma_Z) \tag{4.3.3.4}$$

Since the treatment of Z (made up of the concatenated column vectors of the "work" and "routing" CVs) in this research are asymptotically standardized control variates, then the middle term of equation 4.3.3.2 is eliminated since the mean of $\mathrm{E}[(Z - \mu_Z)] = 0$. Now in the fourth part of this step, the interior-point minimization is evaluated on this Taylor series approximation from equation (4.3.3.1). Finally, to conclude the third step, it is iteratively repeated for both types of radial basis neural network meta-model formulations and through all K cross-validation folds, where 3 statistics, θ_k, are recorded for subsequent analysis:

- $\hat{\theta}^*$: Predicted minimum of selected response
- $\theta^{\text{abs.diff}} = |\hat{\theta}^* - \theta^*|$: Absolute difference between predicted minimum and the true minimum of a selected response

- $\theta^{\text{Norm}} = \text{Norm}\{[\hat{X}^*_{\lambda}\ \hat{X}^*_{\mu(I)}\ \hat{X}^*_{\mu(A)}\ \hat{X}^*_{p(FI)}] - [X^*_{\lambda}\ X^*_{\mu(I)}\ X^*_{\mu(A)}\ X^*_{p(FI)}]\}$: L2 norm distance between the four dimensional set of coordinates for the predicted minimum and the set of coordinates for true minimum for a selected response

The fourth step, illustrated in the bottom green bar of Figure 27, calculates the average for each of the three statistics, $\overline{\theta}_i$, that were recorded in the prior step, across the K independent cross-validation folds. Further, these statistics are evaluated for both types radial basis neural network meta-models. The fourth step concludes by iteratively cycling 100 times back to the second step, to conduct a new random K-fold cross-validation of the same designed experiment, response and number of K cross-validation folds. This step eventually collects 100 bootstrapped statistics, $\overline{\theta}_i$, that are averaged across K folds. This calculation for the ith bootstrapped iteration is shown in the equations directly below.

$$\bar{\theta}^*_i = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}^*_k \quad \text{for } i = 1, 2, \cdots 100 \tag{4.3.3.5}$$

$$\bar{\theta}^{\text{abs.diff}}_i = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}^{\text{abs.diff}}_k \quad \text{for } i = 1, 2, \cdots 100 \tag{4.3.3.6}$$

$$\bar{\theta}^{\text{Norm}}_i = \frac{1}{K} \cdot \sum_{k=1}^{K} \hat{\theta}^{\text{Norm}}_k \quad \text{for } i = 1, 2, \cdots 100 \tag{4.3.3.7}$$

The fifth and final step in this experiment, illustrated in the bottom orange bar of Figure 18, calculates a second average of the statistics for, $\overline{\overline{\theta}}_j$, across the 100 bootstrapped iterations of the previous step, for both types of radial basis neural network meta-models, j.

$$\overline{\overline{\theta}}^*_j = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}^*_i \tag{4.3.3.8}$$

$$\bar{\bar{\theta}}_j^{\text{abs.diff}} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{abs.diff}} \tag{4.3.3.9}$$

$$\bar{\bar{\theta}}_j^{\text{Norm}} = \frac{1}{100} \cdot \sum_{i=1}^{100} \hat{\theta}_i^{\text{Norm}} \tag{4.3.3.10}$$

where

$$j = \{RB_{noCV}, RB_{CV}\}. \tag{4.3.3.11}$$

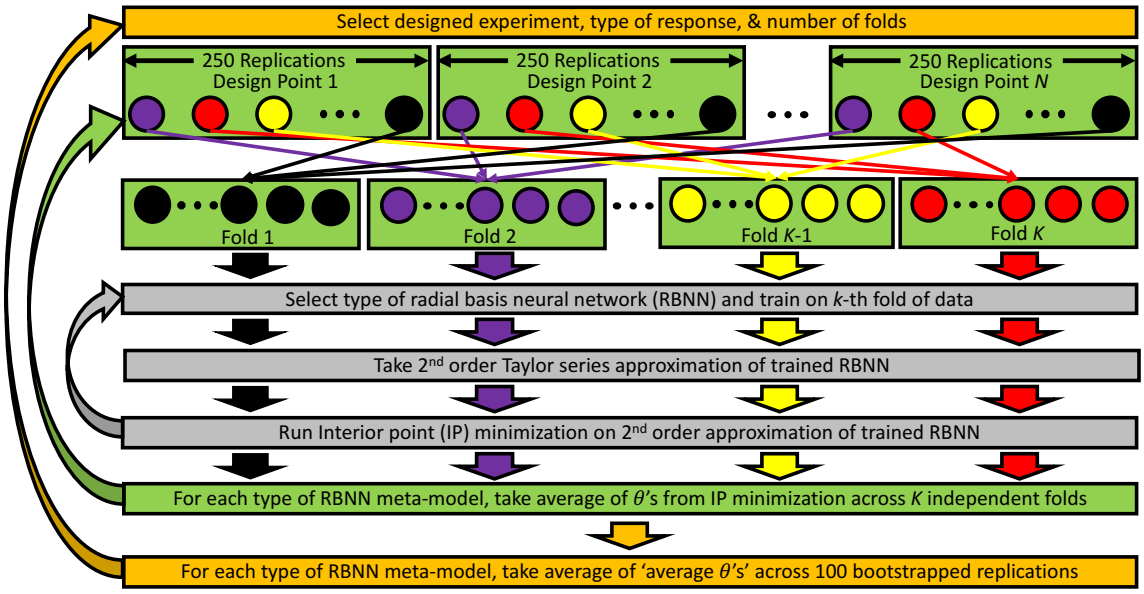


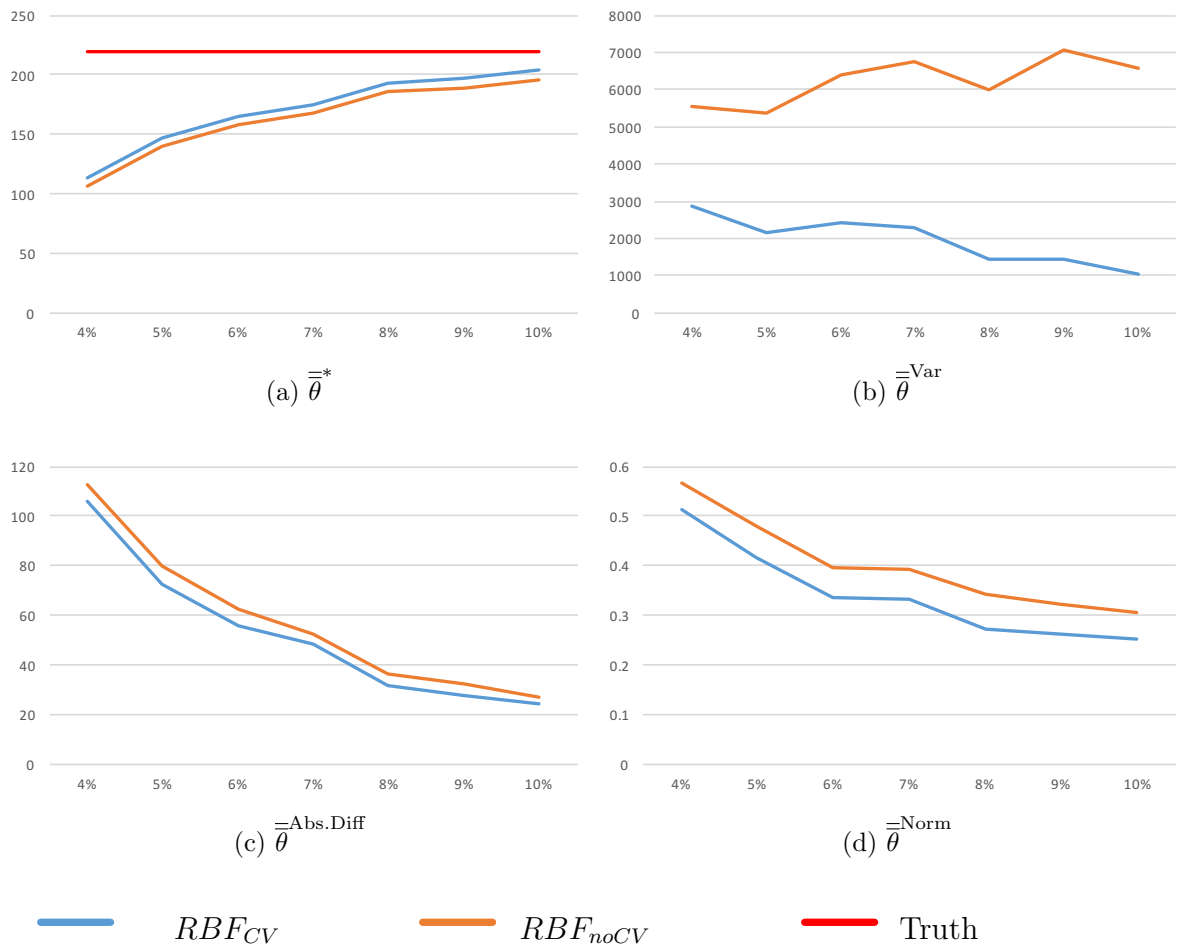
Figure 27. Abstract Flowchart for Radial Basis Neural Network Meta-experiment

4.4 Results

Subsection 4.3, described the methodology (modified from the least squares regression methodology of Subsection 3.3) that was developed to evaluate the performance achieved by extending the statistical framework for radial basis neural networks (see Section 4.2) to include CVs. The results for this meta-experiment demonstrate the

extended framework significantly outperforms the standard formulation. In this section, the results are summarized for the full factorial designed experiment as the type of response and the number of K folds is changed. The performance of the two types of radial basis neural network meta-models, RB_{noCV} and RB_{CV}, are contrasted as these components are interchanged.

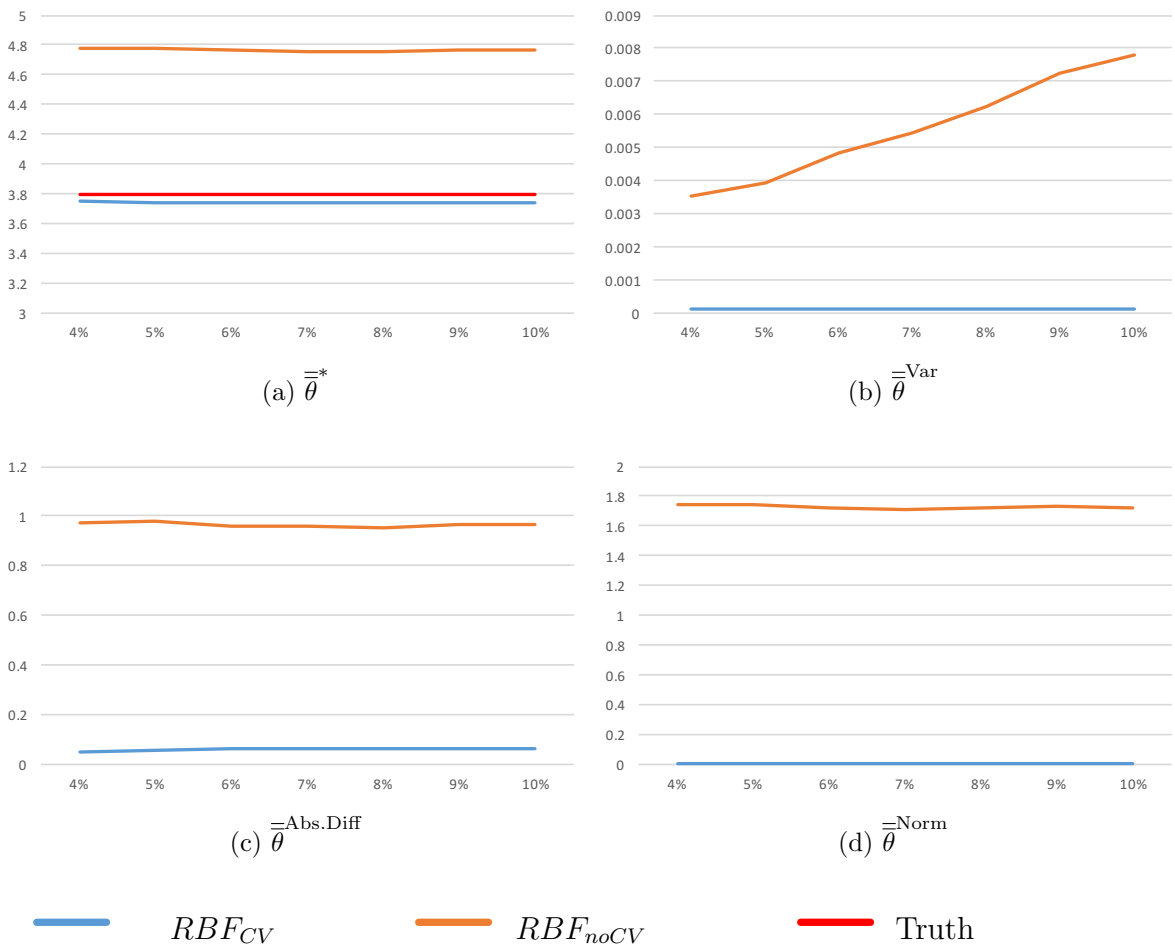
Figures 29 - 30 contrast the performance of two types of radial basis neural network metamodels, RB_{noCV} and RB_{CV}; across a set of metrics for two responses, the average TV sojourn time, Y, and the second nonlinearly transformed cost function, CF_2. Subfigure (a) of Figures 29 - 30, provide the average minimum response, after an interior point optimization routine (see Section 1.2.5, across 100 replications of the number of K folds respective to the number of data points that is in each fold (4% to 10%) and to the type of radial basis neural network meta-model (orange and blue lines) that is utilized. Subfigure also shows the true minimum response (red line), Y^*, evaluated from the Jackson formulas. Subfigure (b) of Figures 29 - 30, provides the average variance of the minimum response observed in Subfigure (a). Subfigure (c) of Figures 29 - 30, provide the average absolute value of the difference between the minimum metamodel response, $\hat{Y}^*$, and the true minimum response, Y^*, that is derived from the Jackson formula analysis. Subfigure (d) of Figures 29 - 30, provides the average $L2$ norm between the meta-model coordinates for $\hat{Y}^*$ and the Jackson formula coordinates for Y^*.



(a) $\overline{\overline{\theta}}^{*}$ (b) $\overline{\overline{\theta}}^{\text{Var}}$

(c) $\overline{\overline{\theta}}^{\text{Abs.Diff}}$ (d) $\overline{\overline{\theta}}^{\text{Norm}}$

Figure 28. Design: FF - Full Factorial, Response: Y - Average TV Sojourn Time



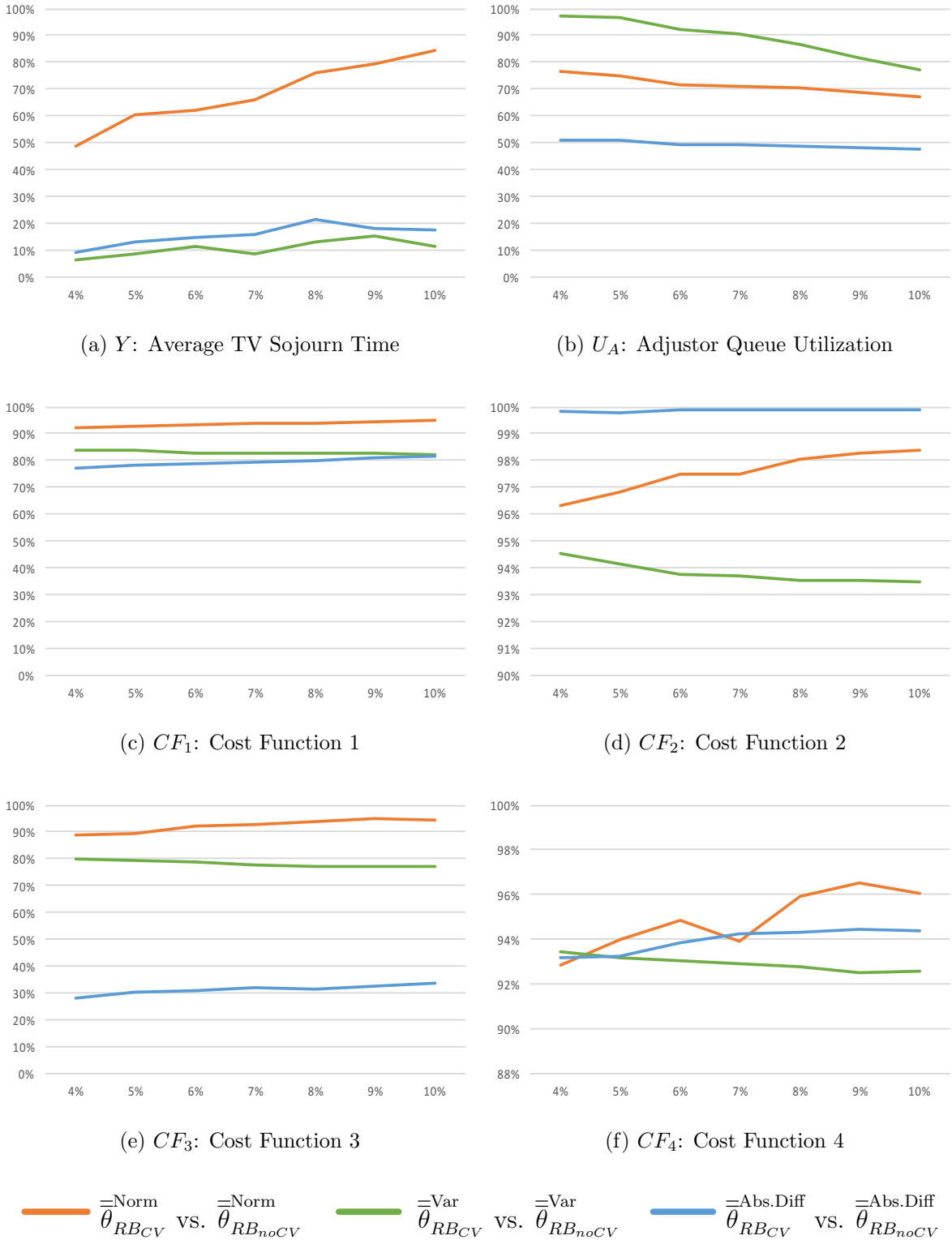
(a) $\bar{\bar{\theta}}^{*}$ (b) $\bar{\bar{\theta}}^{\text{Var}}$

(c) $\bar{\bar{\theta}}^{\text{Abs.Diff}}$ (d) $\bar{\bar{\theta}}^{\text{Norm}}$

Figure 29. Design: FF - Full Factorial, Response: CF_2 - Cost Function 2

Subfigures (a)-(e) of Figure 30, gives the percent of increase or decrease for the three statistics that are illustrated in Figures 28 and 29. For example, the radial basis neural network that includes CVs is contrasted with the the radial basis neural network that excludes CVs for each of the statistics. The three statistics reviewed in these figures give the percent of increase or decrease in the variance of the minimum metamodel response, $\hat{Y}^{*}$, from Subfigure (b) of Figures 28 and 29, the percent of increase or decrease in the absolute difference from Subfigure (c) of Figures 28 and 29, and the percent of increase or decrease in the observed L2 norm from Subfigure (d) of Figures 28 and 29. To compensate for the sheer size of the scope in this research,

only the percent of increase or decrease are presented here. See Appendix A.A.6 for complete set of charts detailing results for all combinations of designs and types of responses.



(a) Y: Average TV Sojourn Time

(b) U_A: Adjustor Queue Utilization

(c) CF_1: Cost Function 1

(d) CF_2: Cost Function 2

(e) CF_3: Cost Function 3

(f) CF_4: Cost Function 4

Figure 30. Design: FF - Full Factorial

By contrasting the meta-models, RB_{noCV} with RB_{CV}, across Figures 28 - 30, shows the inclusion of the asymptotically standardized "work" and "routing" control variates provide a significant degree of improvement to the performance of the radial basis neural network across the selected metrics. See Appendix A.A.6 for comprehensive sets of figures for all other meta-experiments conducted in this study.

4.5 Conclusion

In this chapter, the standard framework for the radial basis neural network that is common in literature was extended for the inclusion of asymptotically standardized "work" and "routing" control variates. An exhaustive heuristic was developed to identify an approximate optimal radial basis neural network architecture for each type of network. A follow-on case study, using the optimal architectures identified by the heuristic, was conducted across 210 meta-experiments that included five designed experiments and six types of responses, which demonstrated the performance of the new extension significantly outperformed the conventional formulation.

V. Conclusion

5.1 Summary

The broader objectives for this research were to develop and test techniques related to meta-models for simulation, variance reduction through control variates, and system dynamic analysis. Missing from the published literature is internalizing higher order interactions of control variates into existing methods used for analyzing simulations. Each of the individual chapters treats this broader research objective within different contexts, to demonstrate the scope of the original contributions.

5.2 Original Contributions

A number of statistical extensions and applications in the field of applied multivariate statistics were accomplished in this research. Each of the contributions are summarized.

5.2.1 Increased Variance Reduction for Direct Estimation of Single Response, Single Population Simulations.

The statistical framework of the controlled unbiased estimation of single response, single population simulations was extended to include second order interactions between the control variates. It is shown that the new extended formulation retains properties associated with unbiased estimators. Through a large case study of a simulation that is characterized by definition of an Jackson open network, we demonstrated that the new extended statistical framework can further increase the amount of variance reduction that is observed beyond the utility of the original first-order framework, provided sufficient replications are conducted. When an exceptional amount of variance was induced into the conditions of the simulated system, the extended

second order statistical framework provided upwards of 3% more variance reduction than what is provided by the standard first order framework.

5.2.2 Unveiling System Dynamics for a Single Response, Single Population Simulations.

The extension of the statistical framework to include second-order interactions of control variates for a single-response, single-population simulation serves two purposes. The first, covered in the previous section, is the realization of additional latent variance reduction. The second purpose of equal, if not greater, value than the first is the clarity provided by evaluating these new second order terms under the lens of interpreting the system dynamics to determine the components in the system that are contributing the most variance. A proposal is made that offers an alternative to the more traditional approach that observes the sensitivity of the simulation response(s) as it is replicated across the input settings for a designed experiment. The new proposal is conducted by observing the sensitivity of the simulation response(s) from the perturbations of the linear and second order interactions of "work" and "routing" CVs. This is an important distinction because the new proposal requires only a single replicated design setting, while the former approach requires a complete designed experiment to evaluate sensitivity. An extensive case study reveals a much stronger understanding of the governing system dynamics for this new proposal, suggesting great promise in practice.

5.2.3 Improved Least Squares Regression Meta-model for Single Response, Multiple Population Simulations.

A new statistical framework to internalize two sets of second order interactions to the existing framework in the literature for least squares regression meta-models

for simulation was developed. The first are the set of interactions between the control variates, and the second set is the interactions between the control variates and designed input variables. Similar to the contributions for variance reduction for direct estimation of single response, single population simulation, this extension does not compromise the unbiased property that holds with the existing framework. A comprehensive case study covering five types of designed experiments and six types of responses, was conducted on a simulation that demonstrated the utility for this new framework, exceeded the performance of the older method that excluded the interactions. A designed region of the simulation, characterized by an exceptional amount of variance that occurred throughout the entire designed area was tested in this new approach. This new framework significantly outperformed the meta-models that were constructed through the old framework in terms of predicting the minimum simulation response in the designed area, reduction of variance around the predicted minimum (from 70-99%), and predicting the optimal design input settings that results in the minimum evaluation of the simulation. A significant improvement of the mean squared error, coefficient of determination (R^2), and Adjusted R^2, across the entire case study was also observed with the new framework.

5.2.4 System Dynamics for Single Response, Multiple Population Simulations.

As a consequence of internalizing the second order interactions of control variates into the least squares regression meta-model formulation, the means to interpret the system dynamics was made available, which strengthened the understanding of the system that was being modeled. The new estimators were shown to explain much more of the system variance that was left unexplained in the original least squares regression meta-model framework. Further, as a direct result of the new

estimators being asymptotically standardized and independent, the new estimates left the coefficients from the original method largely unaffected, proving this method gives much more insight into the system that was being left out by the standard framework.

5.2.5 Improved Radial Basis Neural Network Meta-model for Single Response, Multiple Population Simulations.

A novel approach is presented that extends the statistical framework for the standard radial basis neural network meta-model to accommodate inclusion of control variates as inputs. An extensive case study was conducted, that was similar in scale and scope of the study conducted on least squares regression meta-models. The case study shows this new approach improves the predictive accuracy of the standard radial basis neural network meta-model framework, by orders of magnitude in most cases investigated in the case study. The new method predicted the minimum simulation response and its optimal configuration far more accurately, while observing a significant reduction of variance around the predicted minimum (from 80-99% in most cases).

5.3 Future Research

As outlined in this research, much of the body of literature has largely, overlooked higher order interactions of control variates. In Chapter II, an exceptional amount of variance was induced into the conditions of the simulation to analyze the effect that second order interactions of control variates has on reducing variance. An avenue for future research could investigate third order interactions, if not higher, of a simulation under similar variance conditions.

In Chapter III, a case study was conducted on multiple designed experiments of a simulation with two queues and a single feedback loop that explored a region with a lot of system variance. An interesting aspect left unexplored for this approach is to identify if the proposed extension is scalable to a much larger simulation.

In Chapter IV, the radial basis neural networks framework was manipulated to now include vector of control variates as inputs in addition to the vector of designed input variables. This research showed the control variates under the context of a neural network proves that much higher order interactions of control variates are at play in even relatively small queueing models. It was shown these interaction terms could be used in the regression context to make inferences about the system dynamics of the simulation. An interesting topic would explore how to develop the means to interpret the radial basis neural network framework, by way of control variates, to provide some understanding to the system that is being modeled.

5.4 Final Remarks

The research objectives that were developed in this dissertation were necessary and sufficient steps that can improve the utility of least squares regression and radial basis neural network meta-models for large, complex simulations. The practical application of these contributions can serve as support for present simulation practitioners in their efforts to analyze intractable simulations in environments with limited resources. Further, future research can take the approaches that were investigated in this research in more depth or widen the scope down new avenues.

Appendix A. Additional Tables and Figures

A.1 Information for 73 Simulation Configurations

Table 17. Designed Settings (1-20) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Long Run Mean Responses

	Type of Design					Coded Input Setting				True & Observed Response			
ID#	FF	BB	LHS	CCf	CCc	X_λ	$X_{\mu(I)}$	$X_{\mu(A)}$	$X_{p(FI)}$	U_A	$\bar{U}_A$	Y	$\bar{Y}$
1					✓	2	0	0	0	0.88	0.87	274.90	274.95
2	✓			✓	✓	1	1	1	1	0.94	0.93	519.63	508.83
3	✓			✓	✓	1	1	1	-1	0.90	0.89	358.64	364.87
4		✓				1	1	0	0	0.89	0.88	366.06	364.10
5	✓			✓	✓	1	1	-1	1	0.88	0.88	379.38	377.15
6	✓			✓	✓	1	1	-1	-1	0.84	0.84	298.11	300.74
7			✓			1	0.34	-1	0.14	0.86	0.86	297.63	304.87
8		✓				1	0	1	0	0.92	0.92	365.15	372.21
9		✓				1	0	0	1	0.91	0.90	357.76	344.24
10				✓		1	0	0	0	0.89	0.89	309.39	310.07
11		✓				1	0	0	-1	0.87	0.87	273.86	276.86
12		✓				1	0	-1	0	0.86	0.86	277.12	274.11
13	✓			✓	✓	1	-1	1	1	0.94	0.94	410.86	406.62
14	✓			✓	✓	1	-1	1	-1	0.90	0.90	280.17	283.43
15		✓				1	-1	0	0	0.89	0.89	274.38	272.00
16	✓			✓	✓	1	-1	-1	1	0.88	0.88	270.61	269.53
17	✓			✓	✓	1	-1	-1	-1	0.84	0.84	219.64	218.17
18			✓			0.86	0.61	-0.83	1	0.89	0.88	362.35	366.67
19			✓			0.73	-0.05	0.47	0.83	0.92	0.92	395.98	398.56
20			✓			0.58	-0.64	-0.12	-0.75	0.88	0.87	268.95	269.68

Table 18. Designed Settings (21-40) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses

	Type of Design					Coded Input Setting				True & Observed Response			
ID#	FF	BB	LHS	CCf	CCc	X_λ	$X_{\mu(I)}$	$X_{\mu(A)}$	$X_{p(FI)}$	U_A	$\bar{U}_A$	Y	$\bar{Y}$
21			✓			0.41	0.80	-0.43	0.33	0.89	0.89	386.98	389.68
22			✓			0.34	-1	0.54	-0.40	0.91	0.90	306.76	304.41
23			✓			0.18	1	0.12	-1	0.88	0.88	369.18	369.66
24			✓			0.01	-0.39	0.77	0.64	0.94	0.93	445.69	433.01
25					✓	0	2	0	0	0.90	0.90	626.02	637.45
26		✓				0	1	1	1	0.93	0.93	514.25	515.59
27		✓				0	1	0	1	0.92	0.92	524.88	521.22
28				✓		0	1	0	0	0.90	0.90	438.09	432.48
29		✓				0	1	0	-1	0.88	0.88	376.91	383.24
30		✓				0	1	-1	0	0.87	0.87	397.76	404.33
31					✓	0	0	2	0	0.96	0.95	630.01	596.80
32		✓				0	0	1	1	0.95	0.95	564.14	556.54
33				✓		0	0	1	0	0.93	0.93	431.98	427.25
34		✓				0	0	1	-1	0.91	0.91	357.19	362.23
35					✓	0	0	0	2	0.94	0.94	525.19	529.91
36				✓		0	0	0	1	0.92	0.92	421.16	417.61
37		✓		✓	✓	0	0	0	0	0.90	0.90	355.82	355.36
38				✓		0	0	0	-1	0.88	0.88	309.93	309.49
39					✓	0	0	0	-2	0.86	0.86	275.45	275.64
40		✓				0	0	-1	1	0.89	0.89	359.54	360.54

Table 19. Designed Settings (41-60) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses

	Type of Design					Coded Input Setting				True & Observed Response			
ID#	FF	BB	LHS	CCf	CCc	X_λ	$X_{\mu(I)}$	$X_{\mu(A)}$	$X_{p(FI)}$	U_A	$\bar{U}_A$	Y	$\bar{Y}$
41				✓		0	0	-1	0	0.87	0.87	315.50	316.84
42		✓				0	0	-1	-1	0.85	0.85	281.54	277.85
43					✓	0	0	-2	0	0.84	0.84	290.54	291.14
44		✓				0	-1	1	0	0.93	0.93	385.79	377.08
45		✓				0	-1	0	1	0.92	0.92	366.55	363.74
46				✓		0	-1	0	0	0.90	0.90	309.63	309.31
47		✓				0	-1	0	-1	0.88	0.88	270.29	268.21
48		✓				0	-1	-1	0	0.87	0.87	269.31	272.62
49					✓	0	-2	0	0	0.90	0.90	280.03	284.02
50			✓			-0.15	0.30	0.81	-0.01	0.93	0.93	442.88	449.60
51			✓			-0.27	-0.90	-0.32	-0.31	0.89	0.89	295.33	295.56
52			✓			-0.43	0.52	-0.04	-0.78	0.89	0.89	373.29	376.48
53			✓			-0.54	0.10	-0.75	-0.54	0.88	0.88	333.89	328.75
54			✓			-0.62	-0.23	-0.51	0.42	0.90	0.90	373.03	368.09
55			✓			-0.72	-0.74	0.37	0.59	0.94	0.93	432.07	423.67
56			✓			-0.87	-0.30	1	-0.14	0.94	0.94	476.78	477.86
57	✓			✓	✓	-1	1	1	1	0.97	0.96	949.22	958.42
58	✓			✓	✓	-1	1	1	-1	0.93	0.92	524.80	523.67
59		✓				-1	1	0	0	0.92	0.91	554.49	565.44
60	✓			✓	✓	-1	1	-1	1	0.91	0.90	610.28	576.44

Table 20. Designed Settings (61-73) with Observed (250 simulation replications) & True (Jackson formulas) Long Run Mean Responses

	Type of Design					Coded Input Setting				True & Observed Response			
ID#	FF	BB	LHS	CCf	CCc	X_λ	$X_{\mu(I)}$	$X_{\mu(A)}$	$X_{p(FI)}$	U_A	$\bar{U}_A$	Y	$\bar{Y}$
61	✓			✓	✓	-1	1	-1	-1	0.87	0.86	426.53	425.79
62			✓			-1	0.92	0.23	0.04	0.92	0.92	561.68	554.54
63		✓				-1	0	1	0	0.95	0.94	533.72	538.07
64		✓				-1	0	0	1	0.94	0.93	516.17	513.49
65				✓		-1	0	0	0	0.92	0.91	421.81	429.95
66		✓				-1	0	0	-1	0.90	0.89	359.46	355.58
67		✓				-1	0	-1	0	0.89	0.88	369.53	376.10
68	✓			✓	✓	-1	-1	1	1	0.97	0.96	690.18	687.92
69	✓			✓	✓	-1	-1	1	-1	0.93	0.92	368.83	365.58
70		✓				-1	-1	0	0	0.92	0.91	357.92	362.33
71	✓			✓	✓	-1	-1	-1	1	0.91	0.90	351.24	352.48
72	✓			✓	✓	-1	-1	-1	-1	0.87	0.87	270.57	269.91
73					✓	-2	0	0	0	0.93	0.93	523.30	512.43

A.2 Additional Table for Chapter I: Single Population w/ 2nd Order CVs

Table 21. Average variance reduction relative to standard error of the confidence interval w/o CVs [%]

		Subsets of CVs						
	Sample	W	W	R	WR	WR	WRi	$dWRi$
$X_{p(FI)}$	Size	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$
.05	10		**26**	10		50	9	
.10	10		10	**18**		47	17	
.15	10		5	**19**		39	11	
.20	10		0	**20**		26	-8*	
.05	20		**30**	14		56	50	
.10	20		16	**20**		51	46	
.15	20		12	**21**		45	38	
.20	20		8	**20**		35	26	
.05	40		**32**	16		58	55	38
.10	40		19	**21**		53	51	30
.15	40		15	**22**		48	46	27
.20	40		10	**22**		38	36	3
.05	50		**32**	16		58	56	44
.10	50		19	**21**		54	52	42
.15	50		15	**22**		48	47	36
.20	50		12	**22**		39	38	23
.05	100		**32**	16		59	58	54
.10	100		20	**22**		55	54	51
.15	100		16	**23**		49	49	46
.20	100		13	**22**		39	39	35

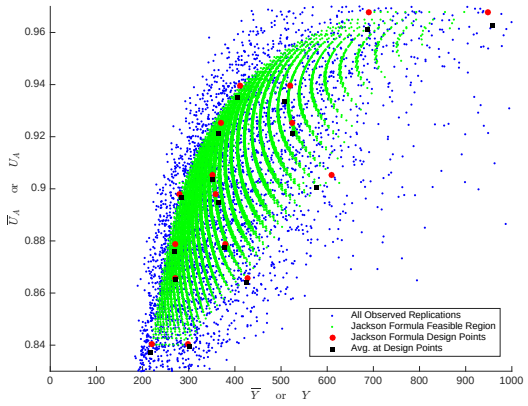
† Bauer & Wilson [16] †† Gibb & Bauer

Table 22. Average variance reduction relative to standard error of the confidence interval w/o CVs [%]

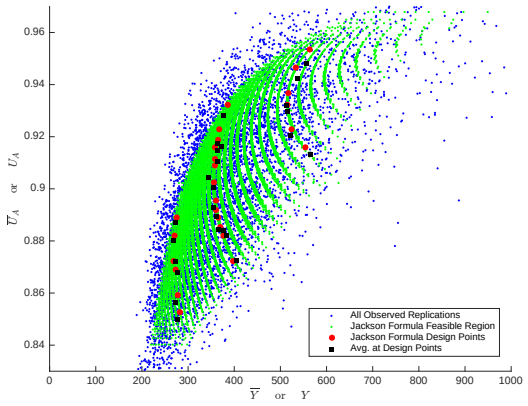
		Subsets of CVs						
	Sample	W	W	R	WR	WR	WRi	$dWRi$
$X_{p(FI)}$	Size	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$B^{\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$	$G^{\dagger\dagger}$
.05	125		**32**	16		59	58	55
.10	125		20	**22**		55	54	53
.15	125		17	**22**		49	49	47
.20	125		14	**22**		40	40	37
.05	200		**32**	16		59	58	58
.10	200		20	**22**		55	55	54
.15	200		17	**23**		50	49	50
.20	200		14	**22**		40	41	40
.05	250		**32**	16		59	59	58
.10	250		20	**23**		55	55	54
.15	250		17	**22**		49	50	50
.20	250		14	**23**		41	41	42
.05	500		**32**	16		59	59	60
.10	500		20	**23**		55	55	56
.15	500		17	**22**		49	49	51
.20	500		14	**23**		41	42	43
.05	1000		**32**	16		59	59	60
.10	1000		20	**23**		55	55	56
.15	1000		17	**22**		49	49	51
.20	1000		14	**23**		41	42	43

† Bauer & Wilson [16] †† Gibb & Bauer

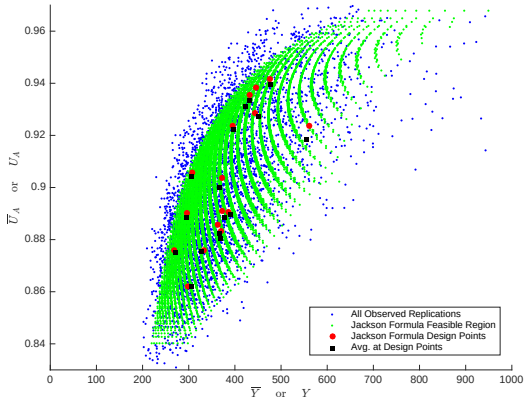
A.3 Additional Figures: Simulation / Jackson Formula Projections



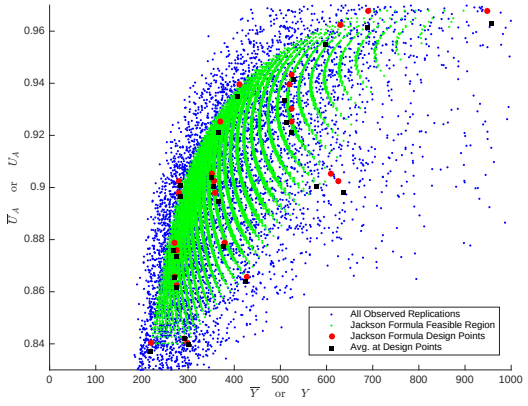
(a) FF: Full Factorial



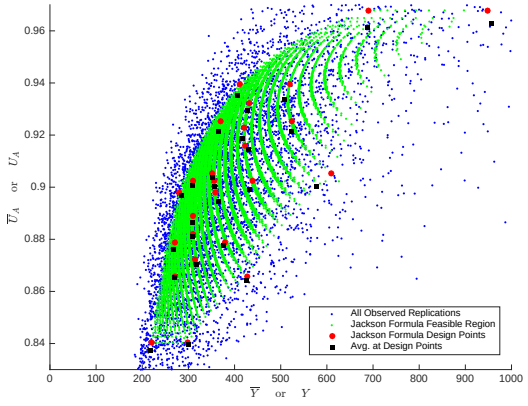
(b) FF: Box-Behnken



(c) LH: Latin Hypercube



(d) CCc: Central Composite - Circumscribed



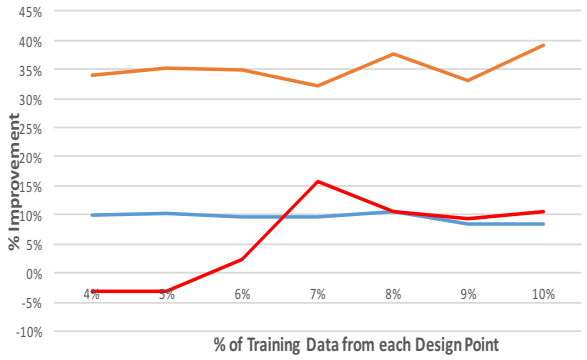
(e) CCf: Central Composite - Faced

Figure 31. Simulation / Jackson Formula Projections for Five Designed Experiment

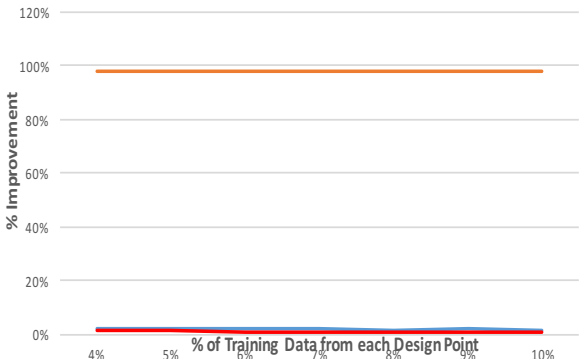
A.4 Estimating Coefficients for $2LSR_{CV}$ meta-model

$X_{p(FI)} =$	4%	5%	6%	7%	8%	9%	10%
0: Intercept	4.71	4.71	4.71	4.71	4.71	4.71	4.71
1: $\boldsymbol{X}_{\lambda}$	0.06	0.06	0.06	0.06	0.06	0.06	0.06
2: $\boldsymbol{X}_{\mu(I)}$	0.09	0.09	0.09	0.09	0.09	0.09	0.09
3: $\boldsymbol{X}_{\mu(A)}$	0.23	0.23	0.23	0.23	0.23	0.23	0.23
4: $\boldsymbol{X}_{\mathrm{p(FI)}}$	0.14	0.14	0.14	0.14	0.14	0.14	0.14
5: $\boldsymbol{W}_{\lambda}$				0.03	0.03	0.03	0.03
6: $\boldsymbol{W}_{\mu(I)}$			0.04	0.04	0.04	0.04	0.04
7: $\boldsymbol{W}_{\mu(A)}$	0.10	0.10	0.10	0.10	0.10	0.10	0.10
8: $\boldsymbol{R}_{\mathrm{p(FI)}}$	0.09	0.09	0.09	0.09	0.09	0.09	0.09
1 x 4	0.06	0.06	0.06	0.06	0.06	0.06	0.06
1 x 5					0.03	0.03	0.03
1 x 8			0.04	0.04	0.04	0.04	0.04
2 x 3	0.08	0.08	0.08	0.08	0.08	0.08	0.08
2 x 4				0.03	0.03	0.03	0.03
2 x 7					0.03	0.03	0.03
3 x 4	0.20	0.20	0.20	0.20	0.20	0.20	0.20
3 x 5	0.05	0.05	0.05	0.05	0.05	0.05	0.05
3 x 6						0.03	0.03
3 x 7	0.11	0.11	0.11	0.11	0.11	0.11	0.11
3 x 8	0.11	0.11	0.11	0.11	0.11	0.11	0.11
4 x 5			0.04	0.04	0.04	0.04	0.04
4 x 7	0.05	0.05	0.05	0.05	0.05	0.05	0.05
4 x 8	0.06	0.06	0.06	0.06	0.06	0.06	0.06
7 x 8					0.03	0.03	0.03

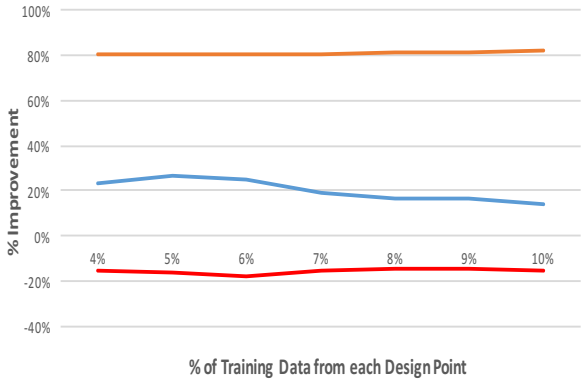
A.5 Chapter III: Least Squares Regression Meta-model Figures



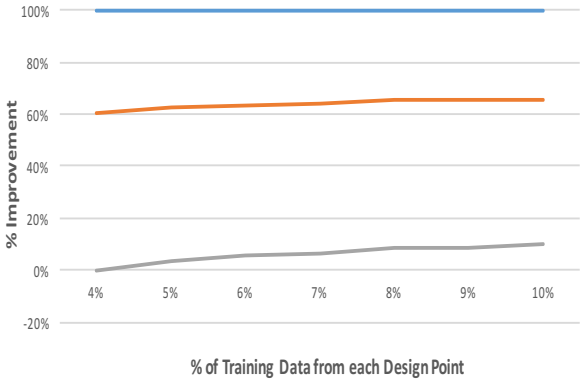
(a) Y: Average TV Sojourn Time



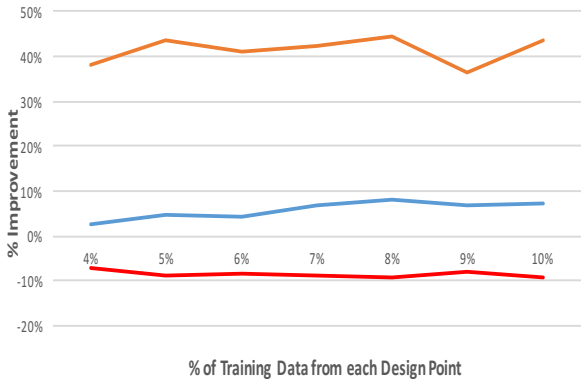
(b) U_A: Adjustor Queue Utilization



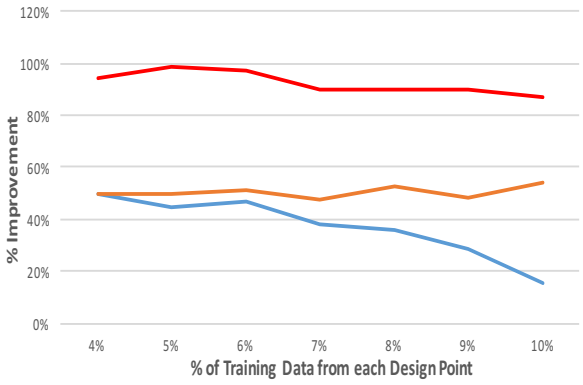
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



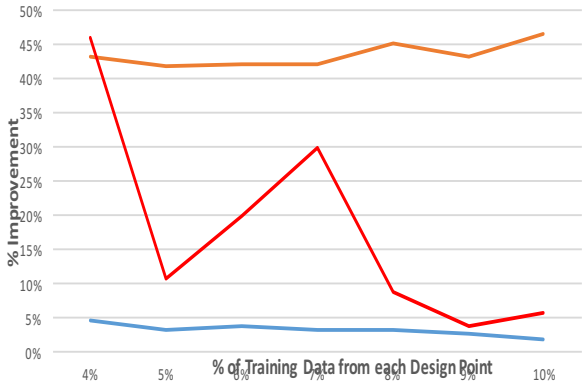
(e) CF_3: Cost Function 3



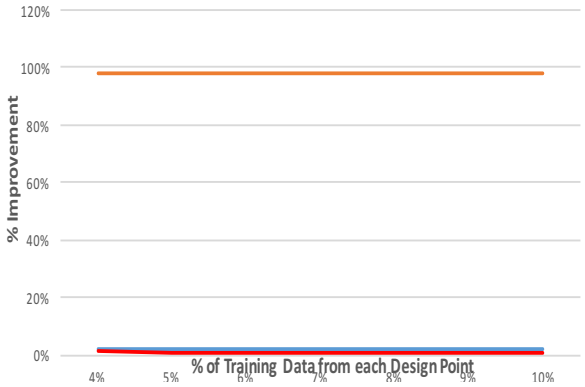
(f) CF_4: Cost Function 4

$\overline{=}$Norm $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Norm $\hat{\theta}_{2LSR_{noCV}}$ $\overline{=}$Var $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Var $\hat{\theta}_{2LSR_{noCV}}$ $\overline{=}$Abs.Diff $\hat{\theta}_{2LSR_{CV}}$ vs. $\overline{=}$Abs.Diff $\hat{\theta}_{2LSR_{noCV}}$

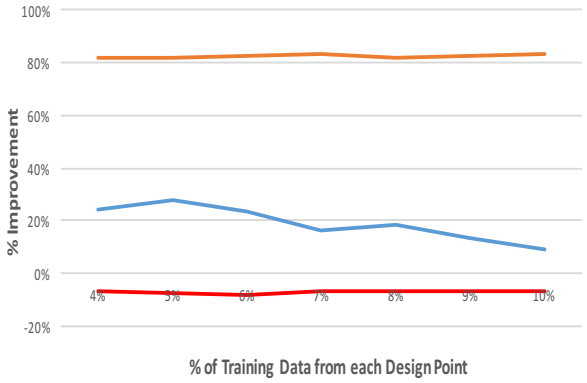
Figure 32. Design: CC_f - Faced Central Composite



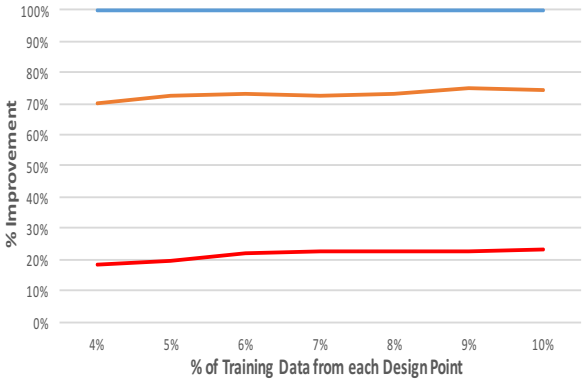
(a) Y: Average TV Sojourn Time



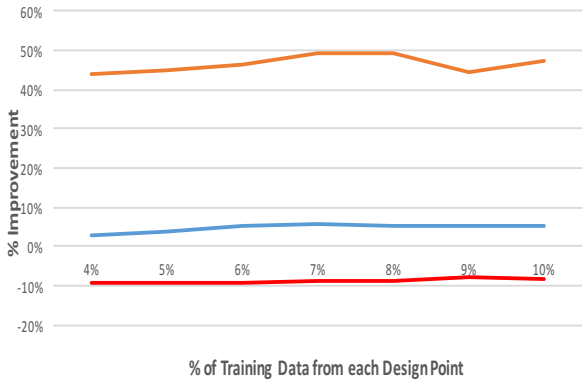
(b) U_A: Adjustor Queue Utilization



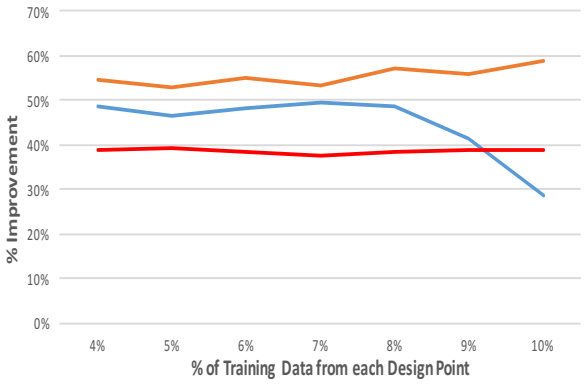
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



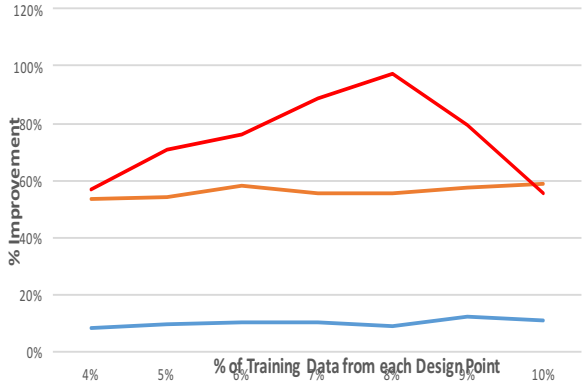
(e) CF_3: Cost Function 3



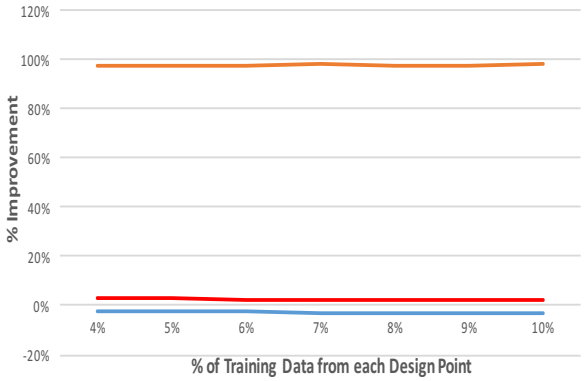
(f) CF_4: Cost Function 4

$\hat{=}$Norm $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Norm $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Var $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Var $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{noCV}}$

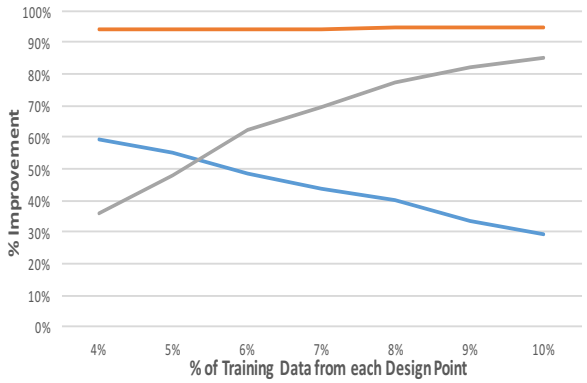
Figure 33. Design: CC_c - Circumscribed Central Composite



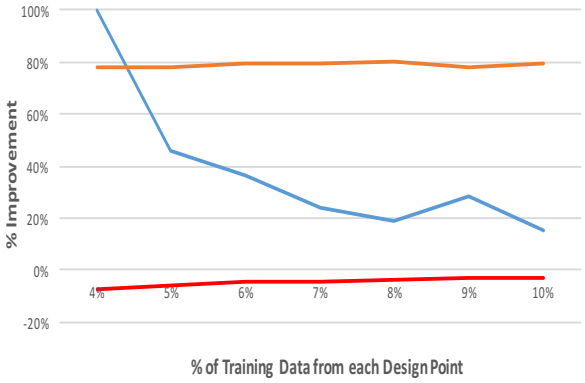
(a) Y: Average TV Sojourn Time



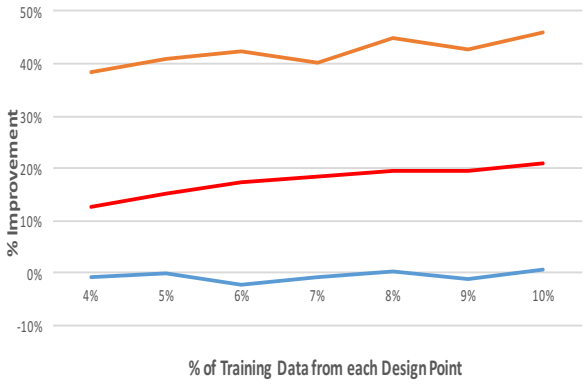
(b) U_A: Adjustor Queue Utilization



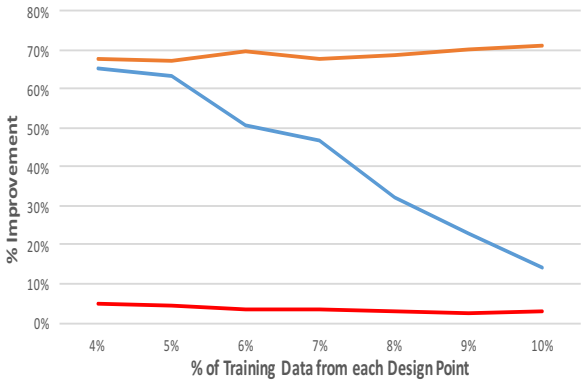
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



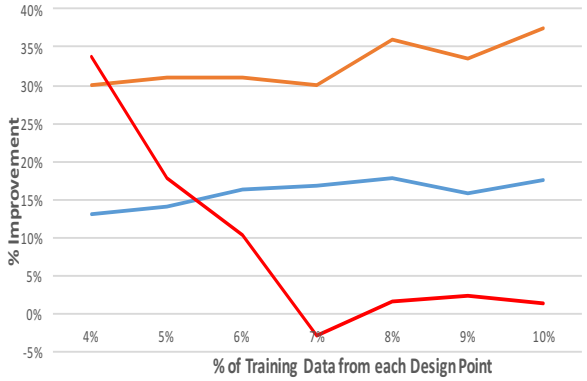
(e) CF_3: Cost Function 3



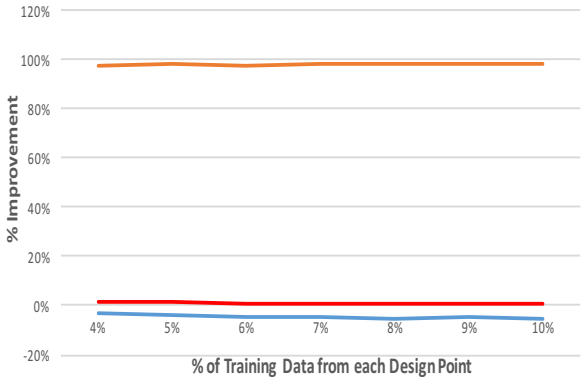
(f) CF_4: Cost Function 4

$\hat{=}$Norm $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Norm $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Var $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Var $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{noCV}}$

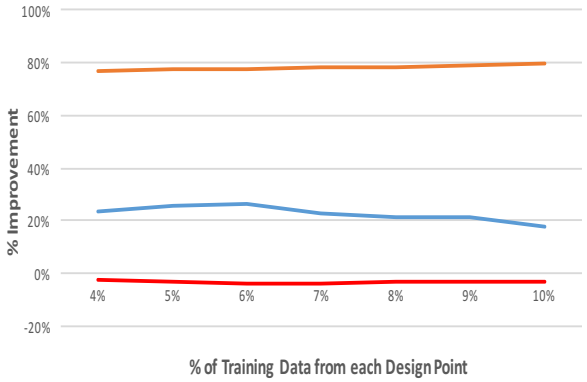
Figure 34. Design: BB - Box-Behnken



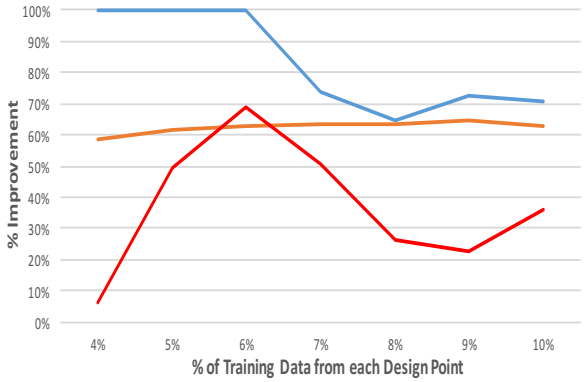
(a) Y: Average TV Sojourn Time



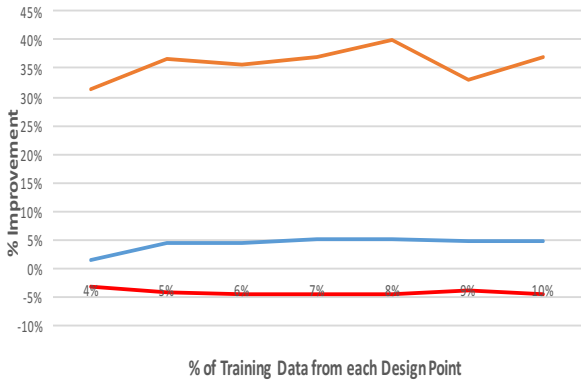
(b) U_A: Adjustor Queue Utilization



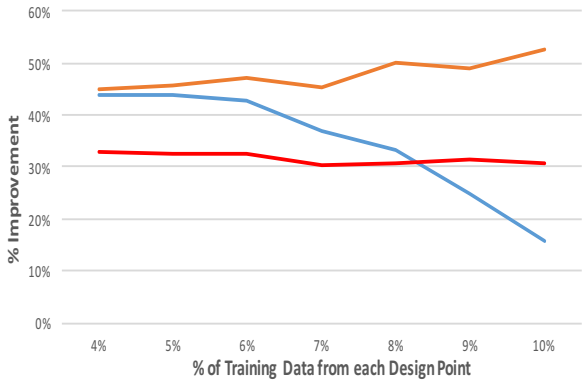
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



(e) CF_3: Cost Function 3

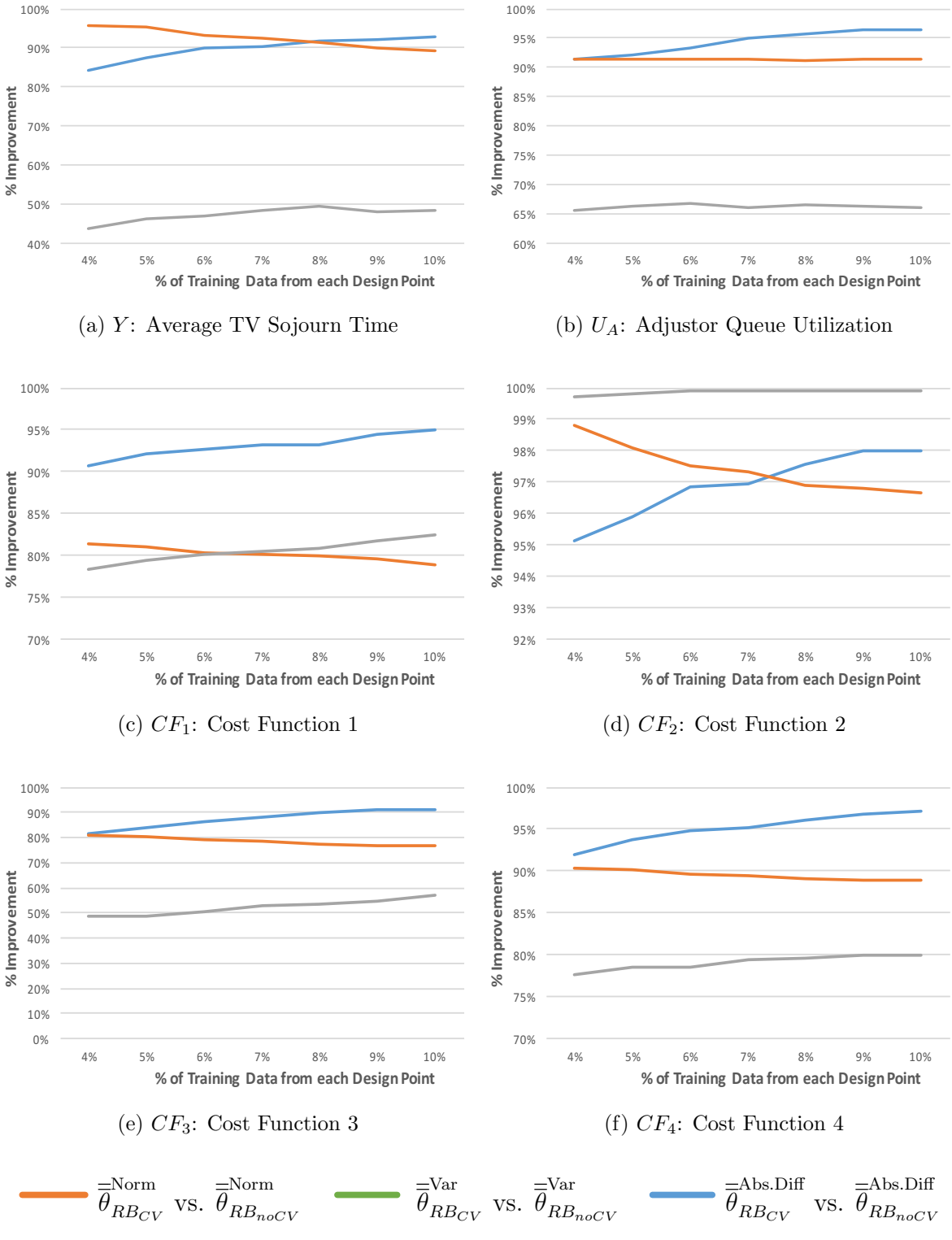


(f) CF_4: Cost Function 4

$\hat{=}$Norm $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Norm $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Var $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Var $\hat{\theta}_{2LSR_{noCV}}$ $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{CV}}$ vs. $\hat{=}$Abs.Diff $\hat{\theta}_{2LSR_{noCV}}$

Figure 35. Design: LH - Latin Hypercube

A.6 Chapter IV: RBNN Meta-model Figures



(a) Y: Average TV Sojourn Time

(b) U_A: Adjustor Queue Utilization

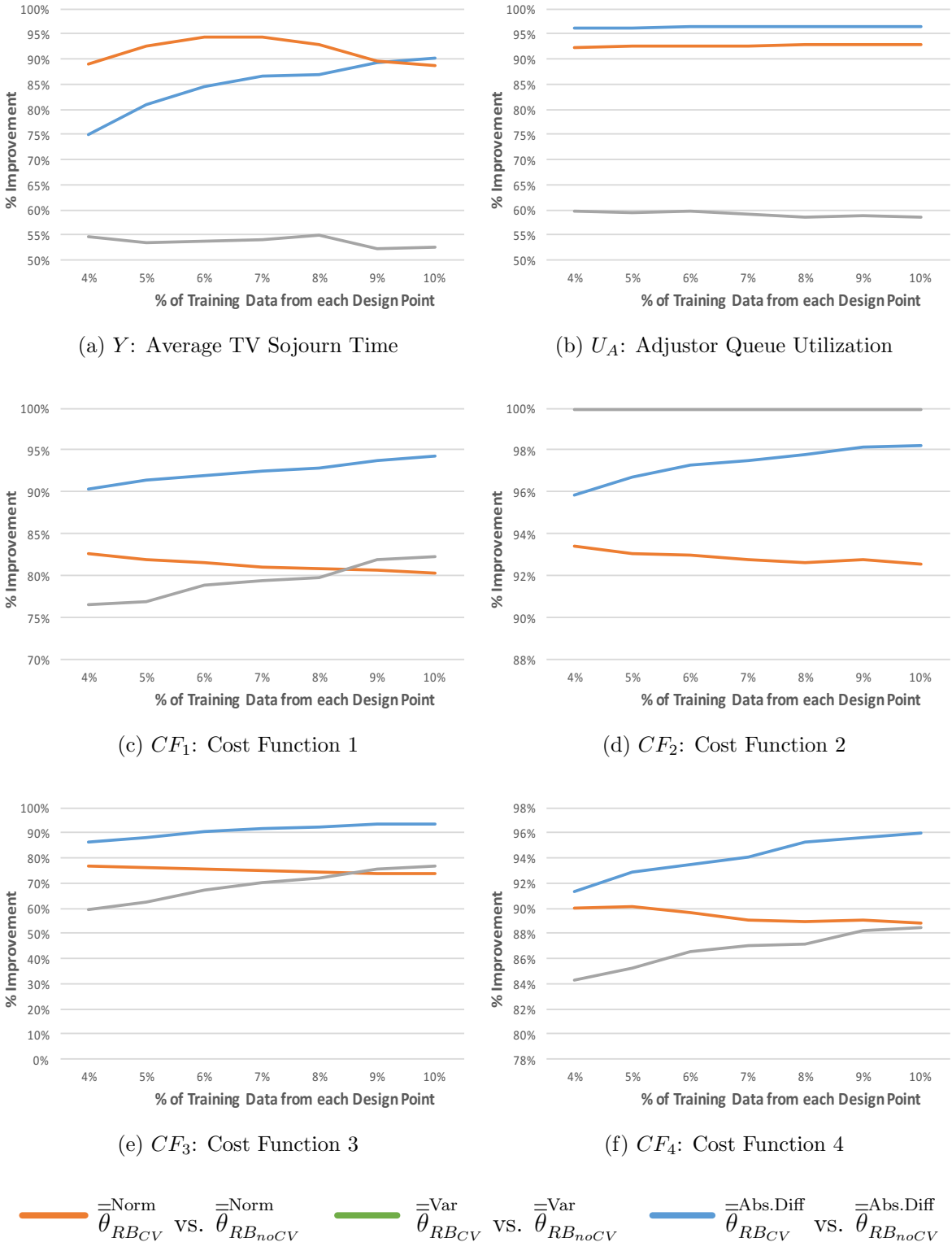
(c) CF_1: Cost Function 1

(d) CF_2: Cost Function 2

(e) CF_3: Cost Function 3

(f) CF_4: Cost Function 4

Figure 36. Design: CC_f - Faced Central Composite



(a) Y: Average TV Sojourn Time

(b) U_A: Adjustor Queue Utilization

(c) CF_1: Cost Function 1

(d) CF_2: Cost Function 2

(e) CF_3: Cost Function 3

(f) CF_4: Cost Function 4

$\bar{=}$Norm $\bar{\theta}_{RB_{CV}}$ vs. $\bar{=}$Norm $\bar{\theta}_{RB_{noCV}}$ $\bar{=}$Var $\bar{\theta}_{RB_{CV}}$ vs. $\bar{=}$Var $\bar{\theta}_{RB_{noCV}}$ $\bar{=}$Abs.Diff $\bar{\theta}_{RB_{CV}}$ vs. $\bar{=}$Abs.Diff $\bar{\theta}_{RB_{noCV}}$

Figure 37. Design: CC_c - Circumscribed Central Composite

(a) Y: Average TV Sojourn Time

(b) U_A: Adjustor Queue Utilization

(c) CF_1: Cost Function 1

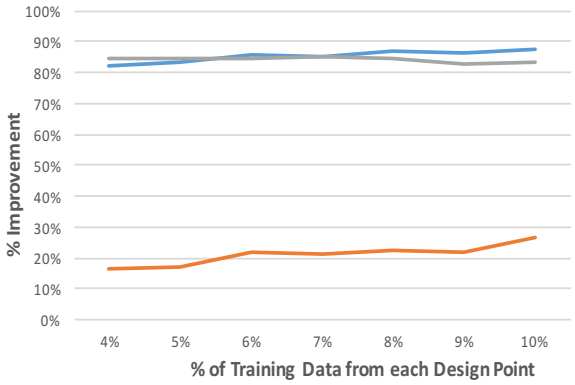
(d) CF_2: Cost Function 2

(e) CF_3: Cost Function 3

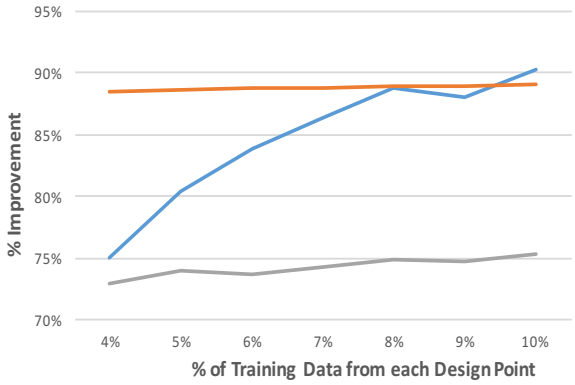
(f) CF_4: Cost Function 4

$\overline{=}^{\text{Norm}}\theta_{RB_{CV}}$ vs. $\overline{=}^{\text{Norm}}\theta_{RB_{noCV}}$ $\overline{=}^{\text{Var}}\theta_{RB_{CV}}$ vs. $\overline{=}^{\text{Var}}\theta_{RB_{noCV}}$ $\overline{=}^{\text{Abs.Diff}}\theta_{RB_{CV}}$ vs. $\overline{=}^{\text{Abs.Diff}}\theta_{RB_{noCV}}$

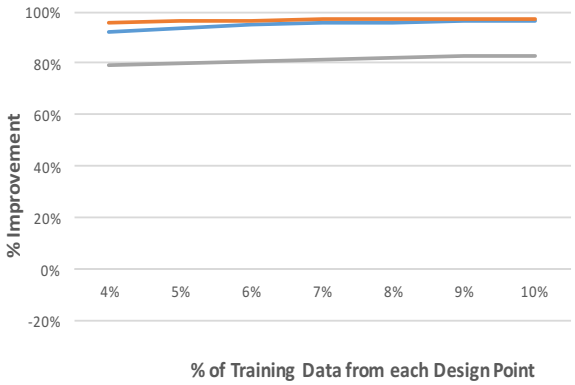
Figure 38. Design: BB - Box-Behnken



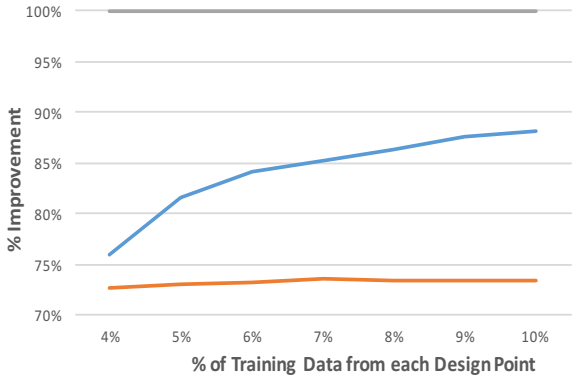
(a) Y: Average TV Sojourn Time



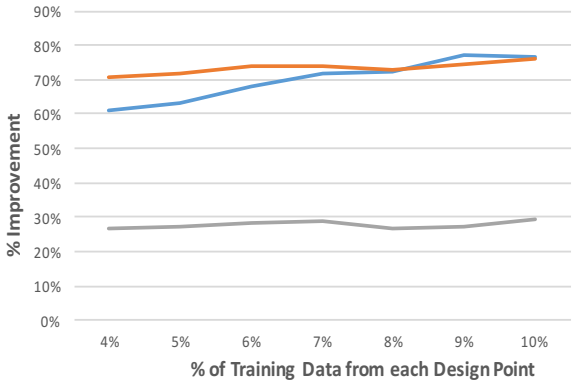
(b) U_A: Adjustor Queue Utilization



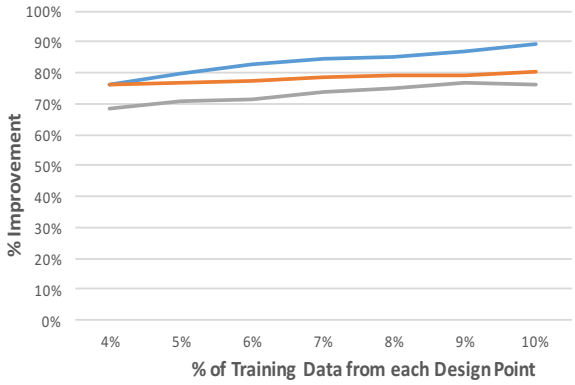
(c) CF_1: Cost Function 1



(d) CF_2: Cost Function 2



(e) CF_3: Cost Function 3



(f) CF_4: Cost Function 4

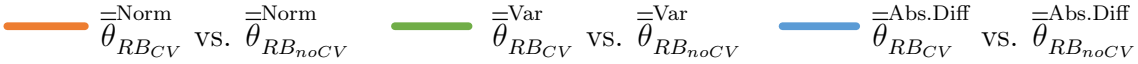


Figure 39. Design: LH - Latin Hypercube

Appendix B. Extended Literature Review

B.1 Single Control Variate (CV) for Single Population, Single Response Simulation

$$E[y(b)] = E[y - b(c - \mu_c)] \tag{B.1.0.1}$$
$$= E[y] - E[b(c - \mu_c)] \tag{B.1.0.2}$$
$$= E[y] - b \cdot E[(c - \mu_c)] \tag{B.1.0.3}$$
$$= E[y] - b \cdot (E[c] - E[\mu_c]) \tag{B.1.0.4}$$
$$= \mu_y - b \cdot (\mu_c - \mu_c) \tag{B.1.0.5}$$
$$= \mu_y - b \cdot (0) \tag{B.1.0.6}$$
$$E[y(b)] = \mu_y \tag{B.1.0.7}$$

It is clear if the term $2b \cdot \text{cov}[y, c]$ is greater than the term $b^2 \cdot \text{var}[c]$, then the right hand side, var$[y]$, has to be greater then the left hand side, var$[y(b)]$, proving Law's assertion [70].

$$\text{var}[y(b)] = \text{var}[y - b \cdot (c - \mu_c)] \tag{B.1.0.8}$$
$$= \text{var}[y - b \cdot c + b \cdot \mu_c] \tag{B.1.0.9}$$
$$= \text{var}[y - b \cdot c] \tag{B.1.0.10}$$
$$\text{var}[y(b)] = \text{var}[y] + b^2 \cdot \text{var}[c] - 2b \cdot \text{cov}[y, c] \tag{B.1.0.11}$$

Law notes [70], the second derivative of B.1.0.11 indicates this function is positive definite, or strictly convex, and therefore b^* must be a global minimizer.

$$\frac{df}{db} = 2b \cdot \text{var}[c] - 2 \cdot \text{cov}[y, c] = 0 \tag{B.1.0.12}$$

$$2b \cdot \text{var}[c] = 2 \cdot \text{cov}[y, c] \tag{B.1.0.13}$$

$$b = \frac{2 \cdot \text{cov}[y, c]}{2 \cdot \text{var}[c]} \tag{B.1.0.14}$$

$$b^* = \frac{\text{cov}[y, c]}{\text{var}[c]} \tag{B.1.0.15}$$

Now substituting the optimal b^* from equation (B.1.0.12) back into equation (B.1.0.11) we can arrive at Kleijnen's [58] conclusion, demonstrated below in equation (B.1.0.16).

$$\text{var}[y(b^*)] = \text{var}[y] + \left[\frac{\text{cov}[y, c]}{\text{var}[c]}\right]^2 \cdot \text{var}[c] - 2\left[\frac{\text{cov}[y, c]}{\text{var}[c]}\right] \cdot \text{cov}[y, c] \tag{B.1.0.16}$$

$$= \text{var}[y] + \left[\frac{\text{cov}[y, c]^2}{\text{var}[c]^2}\right] \cdot \text{var}[c] - 2\left[\frac{\text{cov}[y, c]^2}{\text{var}[c]}\right] \tag{B.1.0.17}$$

$$= \text{var}[y] + \left[\frac{\text{cov}[y, c]^2}{\text{var}[c]}\right] - 2\left[\frac{\text{cov}[y, c]^2}{\text{var}[c]}\right] \tag{B.1.0.18}$$

$$= \text{var}[y] - \frac{\text{cov}[y, c]^2}{\text{var}[c]} \tag{B.1.0.19}$$

$$= \left(1 - \frac{\text{cov}[y, c]^2}{\text{var}[c] \cdot \text{var}[y]}\right) \cdot \text{var}[y] \tag{B.1.0.20}$$

$$\text{var}[y(b^*)] = (1 - \rho_{yc}^2) \cdot \text{var}[y] \tag{B.1.0.21}$$

Now in practice, $\text{cov}[y, c]$ and $\text{var}[c]$ that are found on the RHS of equation (B.1.0.11), are unknown which makes the optimal control coefficient b^*, on the LHS of equation (B.1.0.11), unknown. Porta Nova [101] provides an approach that replaces the RHS of equation (B.1.0.11) with their associated sample statistics, which yields the least squares solution. Bauer notes that under joint normality assumption be-

tween y and c the "least squares solution is also the maximum likelihood solution." Kleijnen [58], gives more detail of the proof of this derivation. Treating the estimation in several methods, including splitting observations or using jackknife method, that all arrive at this same conclusion.

We know from regression theory the least squares estimate for a formulation of the linear regression model can derive the parameter estimates for μ_y and b^*. Content from Johnston & Dinardo [51] is followed in this examination. The bivariate normal distribution, that is assumed here, suggests that conditional expectation for $E[y|c]$ is linear in c and is given as

$$E[y|c] = \mu_y + b^*(c - \bar{c}) \tag{B.1.0.22}$$

Bauer [14] notes that, in this situation the conditional distribution of y given c is also normal:

$$[y|c] \sim N(\mu_y + b^*(c - \bar{c}), \sigma_i^2) \tag{B.1.0.23}$$

For the i^{th} observation, this expectation gives

$$E[y|c_i] = \mu_y + b^*(c_i - \bar{c}), \quad 1 \leq i \leq N, \tag{B.1.0.24}$$

The observed controlled response of the i^{th} observation is denoted by y_i, so we can derive the residual term e_i as

$$\varepsilon_i = y_i - E[y|c_i] = y_i - \mu_y - b^*(c_i - \bar{c}), \quad 1 \leq i \leq N, \tag{B.1.0.25}$$

This residual term represents the influence of all other pieces of the simulation excluding the influence of the control variate $(c - \bar{c})$. With these pieces, we arrive at the bivariate linear regression model of y on c. Given as

$$y_i = \mu_y + b(c_i - \bar{c}) + \varepsilon_i, \quad 1 \leq i \leq N, \tag{B.1.0.26}$$

where the conditional mean of y has two terms: μ_y the parameter to be estimated and the optimal coefficient b^* of the control variate.

$$y_i = \mu_y + b(c_i - \bar{c}) + \varepsilon_i, \quad 1 \leq i \leq N. \tag{B.1.0.27}$$

Now with the normality assumption, the residual term e_i are embodied by

$$\varepsilon_i \sim N(0, \sigma^2). \tag{B.1.0.28}$$

The least squares method will estimate the three parameters of this model, μ_y, b^*, and σ^2. The paired values (μ_y, b^*), are required to fit a specific line. Once this line has been fitted, the residuals associated with that line will be used to evaluate the estimate for σ^2. So let the residuals from any line are given as

$$\varepsilon_i = y_i - \hat{y}_i \tag{B.1.0.29}$$

$$= y_i - \mu_y - b^*(c_i - \bar{c}), \quad 1 \leq i \leq N. \tag{B.1.0.30}$$

Now the residual sum of squares (RSS) can be defined as a function of the paired values (μ_y, b^*) and with the least squares objective one only needs to select the pair that minimizes the RSS given as

$$RSS = \sum_{i=1}^{N} \varepsilon_i^2 = f(\mu_y, b^*), \quad 1 \leq i \leq N. \tag{B.1.0.31}$$

Minimization of this function is characterized by evaluating both partially differentiated equations in terms of each of the paired values at zero. This is given as

$$\frac{\partial(RSS)}{\partial \mu_y} = -2 \sum_{i=1}^{N} \varepsilon_i = 0 \tag{B.1.0.32}$$

$$\frac{\partial(RSS)}{\partial b^*} = -2 \sum_{i=1}^{N} \varepsilon_i (c_i - \bar{c}) = 0 \tag{B.1.0.33}$$

We replace the residual with the values from equation (B.1.0.30) and we get the following,

$$\frac{\partial(RSS)}{\partial \mu_y} = -2 \sum_{i=1}^{N} [y_i - \mu_y - b^*(c_i - \bar{c})] = 0 \tag{B.1.0.34}$$

$$\frac{\partial(RSS)}{\partial b^*} = -2 \sum_{i=1}^{N} [y_i - \mu_y - b^*(c_i - \bar{c})][(c_i - \mu_c)] = 0 \tag{B.1.0.35}$$

From equation (B.1.0.34) and since $\sum_{i=1}^{N}(c_i - \bar{c}) = 0$, we can derive

$$\sum_{i=1}^{N}[y_i - \mu_y] = 0 \tag{B.1.0.36}$$

$$\sum_{i=1}^{N} y_i - \sum_{i=1}^{N} \mu_y = 0 \tag{B.1.0.37}$$

$$\sum_{i=1}^{N} y_i - N\mu_y = 0 \tag{B.1.0.38}$$

$$\sum_{i=1}^{N} y_i = N\mu_y \tag{B.1.0.39}$$

$$\frac{\sum_{i=1}^{N} y_i}{N} = \mu_y \tag{B.1.0.40}$$

$$\bar{y} = \mu_y \tag{B.1.0.41}$$

Substituting equation (B.1.0.41) into equation (B.1.0.35) we get

$$-2\sum_{i=1}^{N}[y_i - \bar{y} - b^*(c_i - \bar{c})][(c_i - \bar{c})] = 0 \tag{B.1.0.42}$$

$$\sum_{i=1}^{N}[y_i - \bar{y} - b^*(c_i - \bar{c})][(c_i - \bar{c})] = 0 \tag{B.1.0.43}$$

$$\sum_{i=1}^{N}(y_i - \bar{y})(c_i - \bar{c}) - b^*\sum_{i=1}^{N}(c_i - \bar{c})^2 = 0 \tag{B.1.0.44}$$

$$b^*\sum_{i=1}^{N}(c_i - \bar{c})^2 = \sum_{i=1}^{N}(y_i - \bar{y})(c_i - \bar{c}) \tag{B.1.0.45}$$

$$b^* = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(c_i - \bar{c})}{\sum_{i=1}^{N}(c_i - \bar{c})^2} \tag{B.1.0.46}$$

Bauer [14] provides the point estimator as

$$\hat{\mu}_y(b^*) = \frac{\sum_{i=1}^{N} y_i(b^*)}{N}. \tag{B.1.0.47}$$

$$c = (1 - \sum_{i=2}^{Q} b_i)c_1 + \sum_{i=2}^{Q} b_i c_i \tag{B.1.0.48}$$

$$= c_1 - c_1 \sum_{i=2}^{Q} b_i + \sum_{i=2}^{Q} b_i c_i \tag{B.1.0.49}$$

$$= c_1 - \sum_{i=2}^{Q} b_i c_1 + \sum_{i=2}^{Q} b_i c_i \tag{B.1.0.50}$$

$$= c_1 + \sum_{i=2}^{Q} (b_i c_i - b_i c_1) \tag{B.1.0.51}$$

$$= c_1 - \sum_{i=2}^{Q} b_i (c_1 - c_i) \tag{B.1.0.52}$$

B.2 Artificial Feed-forward Neural Networks

An artificial neural network is defined as a interconnected mapping of nodes that processes information directs it forward from one layer to the next. The result of the processing and directing of information are outputs that correspond with the inputs and the information that is being extracted. For the purposes of this proposal, the back-propagation method of learning, which modify the weights of processing elements, is the only one treated for the illustration of how neural networks function.

The basic building of the neural network is a neuron, that uses a transfer function to input the weighted sum of its inputs, to output a single value that is directed forward to the next layer of nodes. An arbitrary node can be treated mathematically as

$$y_j = f\left[(\Sigma_i w_{ij}, x_i) + w_{0j}\right], \tag{B.2.0.53}$$

where w_{ij} is the weighted value of the connection from node x_i to node y_j and when $i = 0$ represents the bias that contributes to node y_j, and x_i is the inputted value from node x_i. The function f, which is commonly annotated as an activation function, can take on various forms i.e. hard limiter, threshold logic (as is used by Rosenblatt's perceptron model), hyperbolic tangent, logistic, or linear functions. The important attribute to most of these functions (excluding linear functions) is they can "squash" the input values into a predetermined range. This mathematical treatment of a neuron is illustrated below in Figure 40.

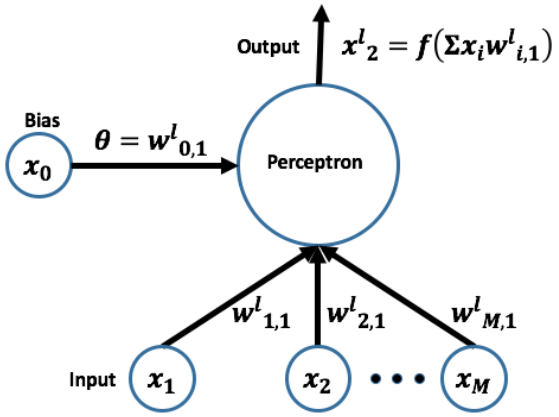


Figure 40. Single perceptron with bias

As was mentioned earlier, these nodes are the basic building blocks of the artificial neural network. When these nodes are connected with other nodes through several layers, they become a framework for the artificial neural network, which in this proposal will be illustrated as a multilayer perceptron architecture, which are known to be perform well at estimating functions or classification problems. A feedforward multilayer perceptron neural network in the format of this proposal has three layers

as is depicted on Figure 41 below that has bias nodes. There are many different architectures beyond just three layers, however having only one hidden layer has been found to be acceptable by Hornik et al. [45] who found that a single hidden layer is necessary to approximate any surface regardless of the degree of nonlinearity that may exist, so long as an adequate number of hidden nodes are invoked in the single layer.

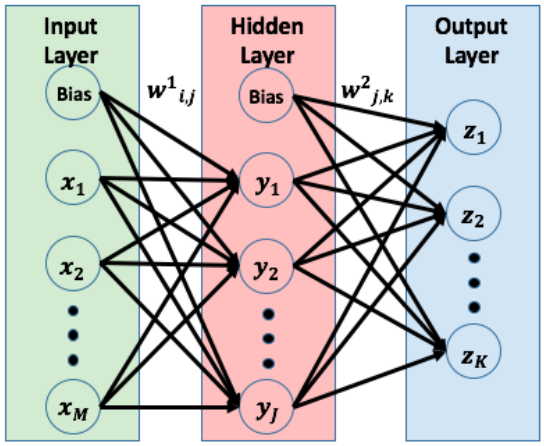


Figure 41. Multilayer perceptron artificial neural network with bias

From this illustration the k^{th} neural network node is given as

$$z_k^n = f(\Sigma_{j=1} w_{j,k}^2 x_j^l), \tag{B.2.0.54}$$

where $w_{i,j}^1$ is the weight for the connection from the input node x_i to the hidden node y_j, $w_{j,k}^2$ is the weight for the connection from the hidden node y_j to the output node z_k, x_0^l is the bias term for the hidden layer, y_j^l is the output of hidden node j, x_i^l is the output of the input node i, M is the number of nodes in the input layer, J is the number of nodes in the hidden layer, and K is the number of nodes in the output layer.

The properties for each layer contributes to the strengths of this metamodel method. The input layer of nodes that is highlighted in green from Figure 41 receives data from an outside source processes them in some manner in many cases using a linear activation function. The hidden layer of nodes that is highlighted in red from Figure 41 receives the processed information from the input layer, and pushes them through a second transformation which are commonly the specialized sigmoids for treating any nonlinearities of the data. The out layer of nodes that is highlighted in blue from Figure 41 receives the processed information from the hidden layer and outputs this information as results of the network. These outputs are commonly annotated as "targets" and used for training the network to adjust these targets so that it is capable to predict the actual true values of the system being modeled by the neural network.

The training process that is illustrated here to demonstrate the capabilities of the neural network is focused on the back-propagation which is the most common, although there is many types of methods that exist today. As was mentioned in the previous section, this method was first introduced by Werbos [125] and was considered mainstream with Rumelhart et al. [110].

In the first phase of the neural network modeling process, a vector of input data is presented to the network to the input nodes of the network, and then processed forward through the layers of the network until it reaches the output layer and recorded in a new vector of network response values. The second phase of this process seeks to calculate the sum of squared error between the "targets" and the neural network response and minimize it in an effort to increase the predictive accuracy of the neural network. The sum of squared error is given as

$$SSE = \sum_{k=1}^{K} \frac{(t_{k,p} - z_{k,p})^2}{2}, \tag{B.2.0.55}$$

where $t_{k,m}$ is the target output for the p^{th} exemplar, $z_{k,p}$ is the approximated output for output node k for the p^{th} exemplar. It should be noted that the sum of squared error may be replaced with the respective mean squared error calculation given as

$$MSE = \frac{SSE}{K}, \tag{B.2.0.56}$$

however the overall results of either algorithm should not differ according to Greene [38]. These squared error formulations are used for what is called "instantaneous" error calculations that derive the partial derivatives based on the presentation of a single exemplar. In the case where the entire training set of exemplars is presented, the calculation for total squared error is used and is given as

$$TSE = \sum_{p}^{P} \sum_{k}^{K} \frac{(t_{kp} - z_{kp})^2}{2}, \tag{B.2.0.57}$$

where $t_{k,m}$ is the target output for the p^{th} exemplar, $z_{k,p}$ is the approximated output for output node k for the p^{th} exemplar.

The gradient search of the error surface adjusts the connection weights by the product of the differentiated surface relative to the connection weights (which provides the search direction) with a predetermined learning rate value (which provides the distance to search in the direction). Rumelhart et al. [110] provided the break-

through that made this gradient error search method possible with the introduction of continuous activation functions.

The mathematical treatment of the updated weights is given as

$$W^{+} = W^{-} - \eta \cdot \frac{\partial SSE}{\partial W}, \tag{B.2.0.58}$$

where W^{+} is the new matrix of weights, W^{-} is the old matrix of weights, η is the learning rate (given to be $0 < \eta < 1$, and $\frac{\partial SSE}{\partial W}$ is a matrix of partials in relation to the weights of each layers and each connection.

It is important to note here that this method can have difficulties being trapped in local minimum found within domain of the error surface, which can ultimately prevent the network from discovering the global minimum. This convergence to local minima can be overcome with an additional term that is called "momentum". This new term that is appended to the learning rate multiplier allows for the network to push through local minimum, however, if given too much weight it will make the system untenable which will not fix the problem of finding the global minimum. The back-propagation algorithm given in equation (B.2.0.58) is now given as

$$W^{+} = W^{-} - \eta \cdot \frac{\partial SSE}{\partial W} + m \cdot (W^{-} - W^{--}), \tag{B.2.0.59}$$

where W^{+} is the new matrix of weights, W^{-} is the old matrix of weights, η is the learning rate given to be $0 < \eta < 1$, $\frac{\partial SSE}{\partial W}$ is a matrix of partials in relation to the weights of each layers and each connection, m is a constant given to be $0 < m < 1$, and W^{--} is a weight matrix that is one iteration older than W^{-}.

An additional challenge for neural networks is based on feature selection theory. This deal with filtering out input factors that do not contribute towards the accuracy of a neural network, keeping only the most salient factors. If the noisy factors are retained, the neural network will continue to train the network until it is overfit to redundant factors. Bauer et al. [13] provided a method that uses a signal-to-noise ratio that measures all input factors to an additional arbitrary noise factor that is introduced into the model. The factors with the highest ratios in theory provide the strongest signal to contributing to the goodness of fit of the neural network and therefore kept, while those that contribute nothing more than the arbitrary noise factor is excluding from entering the neural network. This method iteratively makes the ratio calculation, throws out the factor with the lowest ratio, and restarts the process until all factors have been thrown out, while with each iteration the sum of squared error is recorded. At the end of this process, the combination of factors that produced the lowest squared error will be kept and used for further analytic efforts.

Finally, it is common practice for all input factors to be pre-processed into standardized, normalized, or coded (depending on personal preference) versions of the input factors. The "unit-less" standardized method which provides for a zero mean and unit variance is given as

$$x'_{p,m} = \frac{x_{p,m} - \bar{x}_m}{S_m} \tag{B.2.0.60}$$

where $x'_{p,m}$ is the standardized p^{th} exemplar for the m^{th} input factor, $x_{p,m}$ is the unadjusted p^{th} exemplar for the m^{th} input factor, $\bar{x}_m$ is the observed sample mean for the m^{th} input factor x_m, and S_m is the observed sample standard deviation for the m^{th} input factor x_m.

The observed sample mean, $\bar{x}_m$, for the m^{th} input factor x_m is given as

$$\bar{x}_m = \frac{\sum_{p=1}^{P} x_{p,m}}{P}. \tag{B.2.0.61}$$

The observed sample standard deviation S_m for the m^{th} input factor x_m is given as

$$S_m = \sqrt{\frac{\sum_{p=1}^{P}(x_{p,m} - \bar{x}_m)^2}{P-1}} \tag{B.2.0.62}$$

The normalized method which provides a range between 0 and 1 is given as

$$x'_{p,m} = \frac{x_{p,m} - \min(x_m)}{\max(x_m) - \min(x_m)}, \tag{B.2.0.63}$$

where $x'_{p,m}$ is the normalized p^{th} exemplar for the m^{th} input factor, $x_{p,m}$ is the unadjusted p^{th} exemplar for the m^{th} input factor, $\min(x_m)$ is the minimum observed value of the m^{th} input factor x_m, and $\max(x_m)$ is the maximum observed value of the m^{th} input factor x_m.

The coding method which provides a range between -1 and 1 is given as

$$x'_{p,m} = \frac{x_{p,m} - \frac{\max(x_m) - \min(x_m)}{2}}{\frac{\max(x_m) - \min(x_m)}{2}}, \tag{B.2.0.64}$$

where $x'_{p,m}$ is the coded p^{th} exemplar for the m^{th} input factor, $x_{p,m}$ is the unadjusted p^{th} exemplar for the m^{th} input factor, $\min(x_m)$ is the minimum observed value of the m^{th} input factor x_m, and $\max(x_m)$ is the maximum observed value of the m^{th} input factor x_m.

All of these methods describes above ensures the large valued input factors will not dominant over other input factors that have much smaller valued spreads in the process of training the neural network.

B.3 Least Squares Optimality

$$\begin{aligned}
\text{SSE} &= \sum \varepsilon^2 && \text{(B.3.0.65)}\\
&= \varepsilon'\varepsilon && \text{(B.3.0.66)}\\
&= (Y - X\beta)'(Y - X\beta) && \text{(B.3.0.67)}\\
&= Y'Y - Y'X\beta - \beta'X'Y + \beta'X'X\beta && \text{(B.3.0.68)}\\
&= Y'Y - 2\beta'X'Y + \beta'X'X\beta && \text{(B.3.0.69)}
\end{aligned}$$

$$\begin{aligned}
-2X'Y + 2X'X\beta &= 0 && \text{(B.3.0.70)}\\
2X'X\beta &= 2X'Y && \text{(B.3.0.71)}\\
X'X\beta &= X'Y && \text{(B.3.0.72)}\\
\beta &= (X'X)^{-1}X'Y && \text{(B.3.0.73)}
\end{aligned}$$

B.4 Unbiased Quadratic Control Variates

This section will clarify the distinctions of the control variate methods proposed by Moy [82, 83], Radema [105] and reviewed by Kleijnen [57] , and the formulation attributed to them by Nelson [87]. They first define the trivial situation involving a univariate response, y, with multiple control variate estimators

$$y_{cm} = y - \sum_{k=1}^{K} \beta_k \cdot (\bar{c}_k - \mu_{c_k}) \tag{B.4.0.74}$$

where the average output of the kth stochastic process $\bar{y}_k$ is defined as

$$\bar{c}_k = \sum_{t=1}^{M_k} \frac{C_{kt}}{M_k}, \tag{B.4.0.75}$$

y_{kt} is the tth instance of the kth stochastic process in the simulation, μ_{yk}, is the true mean programmed for the kth stochastic process, y_{cm}, is the controlled response for the simulation, y, is the uncontrolled response for the simulation, and β_k, are associated coefficients.

Moy [82, 83] generalized the method above for the case of a single stochastic process,

$$y_{cm} = y - \beta \cdot (\bar{c} - \mu_c), \tag{B.4.0.76}$$

by replacing the evaluated average $\bar{c}$ with some arbitrary function, g. This function g takes as input, M instances of c_t, that are generated from a stochastic process internal to a simulation. Kleijnen [57] acknowledges the expected value of g must be known for this method to hold. Moy [82, 83] characterizes this control variate as,

$$y_{cg} = y - g(c_1, c_2, \ldots, c_M) + \mathrm{E}\{g(c_1, c_2, \ldots, c_M)\}, \tag{B.4.0.77}$$

where $g(c_1, c_2, \ldots, c_M)$ is the arbitrary function, y, is the uncontrolled simulation response of interest, y_{cg}, is the generalized form of the controlled simulation responses, $\mathrm{E}\{g(c_1, c_2, \ldots, c_M)\}$, is known. When the expected value transformation is passed through this formulation it is revealed to retain its properties as an unbiased estimator, $\mathrm{E}\{y_{cg}\} = \mathrm{E}\{y\}$. Kleijnen [57] annotates the CV, $g(c_1, c_2, \ldots, c_M)$, must have a "strong" positive correlation with the response. When considering the variance formulation of y_{cg},

$$\sigma^2\{y_{cg}\} = \sigma^2\{y\} + \sigma^2\{g(c_1, c_2, \ldots, c_M)\} - 2 \cdot \sigma\{x, g(c_1, c_2, \ldots, c_M)\}, \quad \text{(B.4.0.78)}$$

to induce variance reduction, $\sigma^2\{y_{cg}\} < \sigma^2\{y\}$, specifically requires the covariance of the generalized CV with the simulation response be greater than half of the variance of the simulation response,

$$2 \cdot \sigma\{x, g(c_1, c_2, \ldots, c_M)\} > \sigma^2\{g(c_1, c_2, \ldots, c_M)\}. \quad \text{(B.4.0.79)}$$

Moy [82, 83] provides two examples for this generalized formulation, a linear and quadratic polynomial function. Moy [82, 83] formulates the linear polynomial CV as,

$$y_{cl} = y - g(c_1, c_2, \ldots, c_M) + \mathrm{E}\{g(c_1, c_2, \ldots, c_M)\}, \quad \text{(B.4.0.80)}$$

where

$$g(c_1, c_2, \ldots, c_M) = b_1 + b_2 \cdot \sum_{t=1}^{M} c_t. \tag{B.4.0.81}$$

This formulation for y_{cl} is shown below to resolve to the original CV formulation, y_c, based on the average with β coefficient.

$$\begin{aligned}
y_{gl} &= y - g(c_1, c_2, \ldots, c_M) + \mathrm{E}\{g(c_1, c_2, \ldots, c_M)\} && \text{(B.4.0.82)}\\
&= y - \left[b_1 + b_2 \cdot \sum_{t=1}^{M} c_t\right] + \mathrm{E}\left\{\left[b_1 + b_2 \cdot \sum_{t=1}^{M} c_t\right]\right\} && \text{(B.4.0.83)}\\
&= y - b_1 - b_2 \cdot \sum_{t=1}^{M} c_t + \mathrm{E}\{b_1\} + \mathrm{E}\left\{b_2 \cdot \sum_{t=1}^{M} c_t\right\} && \text{(B.4.0.84)}\\
&= y - b_1 - b_2 \cdot \sum_{t=1}^{M} c_t + b_1 + b_2 \cdot \mathrm{E}\left\{\sum_{t=1}^{M} c_t\right\} && \text{(B.4.0.85)}\\
&= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - \mathrm{E}\left\{\sum_{t=1}^{M} c_t\right\}\right) && \text{(B.4.0.86)}\\
&= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - \sum_{t=1}^{M} \mathrm{E}\{c_t\}\right) && \text{(B.4.0.87)}\\
&= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - M \cdot \mu_c\right) && \text{(B.4.0.88)}\\
&= y - b_2 \cdot \left(M \cdot \left(\frac{1}{M} \cdot \sum_{t=1}^{M} c_t - \mu_c\right)\right) && \text{(B.4.0.89)}\\
&= y - b_2 \cdot M \cdot \left(\sum_{t=1}^{M} c_t - \mu_c\right) && \text{(B.4.0.90)}\\
&= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) && \text{(B.4.0.91)}
\end{aligned}$$

Letting $\beta = b_2 \cdot M$, then arrive to original CV formulation

$$y_c = y - \beta \cdot (\bar{c} - \mu_c). \tag{B.4.0.92}$$

Before going into detail on Moy's quadratic formulation, the second order polynomial expansion of the standard formulation of the CV, $(\bar{c} - \mu_c)$, is given as

$$\begin{aligned}
y_c &= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot (\bar{c} - \mu_c)^2 && \text{(B.4.0.93)}\\
&= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot (\bar{c} - \mu_c) \cdot (\bar{c} - \mu_c) && \text{(B.4.0.94)}\\
&= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot \left[(\bar{c})^2 - 2 \cdot \bar{c} \cdot \mu_c + \mu_c^2\right] && \text{(B.4.0.95)}\\
&= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot \left[(\bar{c})^2 - 2 \cdot \bar{c} \cdot \mu_c + \mu_c^2\right] && \text{(B.4.0.96)}\\
&= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot \left[(\bar{c})^2 + \mu_c^2 - 2 \cdot \bar{c} \cdot \mu_c\right] && \text{(B.4.0.97)}\\
&= y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot \left[(\bar{c})^2 - (\mu_c^2 + 2 \cdot \bar{c} \cdot \mu_c)\right] && \text{(B.4.0.98)}
\end{aligned}$$

Passing the expected value function through equation (B.4.0.98) is shown below,

$$\begin{aligned}
\mathrm{E}\{y_c\} &= \mathrm{E}\{y - \beta_1 \cdot (\bar{c} - \mu_c) - \beta_2 \cdot (\bar{c} - \mu_c)^2\} && \text{(B.4.0.99)}\\
&= \mathrm{E}\{y\} - \mathrm{E}\{\beta_1 \cdot (\bar{c} - \mu_c)\} - \mathrm{E}\{\beta_2 \cdot (\bar{c} - \mu_c) \cdot (\bar{c} - \mu_c)\} && \text{(B.4.0.100)}\\
&= \mathrm{E}\{y\} - \beta_1 \cdot \mathrm{E}\{(\bar{c} - \mu_c)\} - \beta_2 \cdot \mathrm{E}\{(\bar{c} - \mu_c) \cdot (\bar{c} - \mu_c)\} && \text{(B.4.0.101)}\\
&= \mathrm{E}\{y\} - \beta_1 \cdot \mathrm{E}\{(\bar{c} - \mu_c)\} - \beta_2 \cdot \mathrm{E}\{[(\bar{c})^2 - 2 \cdot \bar{c} \cdot \mu_c + \mu_c^2]\} && \text{(B.4.0.102)}\\
&= \mathrm{E}\{y\} - \beta_1 \cdot (\mathrm{E}\{\bar{c}\} - \mathrm{E}\{\mu_c\}) - \beta_2 \cdot [\mathrm{E}\{(\bar{c})^2\} - \mathrm{E}\{2 \cdot \bar{c} \cdot \mu_c\} + \mathrm{E}\{\mu_c^2\}] && \text{(B.4.0.103)}\\
&= \mathrm{E}\{y\} - \beta_1 \cdot (\mu_c - \mu_c) - \beta_2 \cdot [\mathrm{E}\{(\bar{c} \cdot \bar{c})\} - 2 \cdot \mu_c \cdot \mathrm{E}\{\bar{c}\} + \mu_c^2] && \text{(B.4.0.104)}
\end{aligned}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\mathrm{E}\left\{ \sum_{t=1}^{M} \sum_{s=1}^{M} \frac{c_t}{M} \cdot \frac{c_s}{M} \right\} - 2 \cdot \mu_c \cdot \mu_c + \mu_c^2 \right] \quad \text{(B.4.0.105)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \mathrm{E}\left\{ (\sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s) \right\} - 2 \cdot \mu_c^2 + \mu_c^2 \right] \quad \text{(B.4.0.106)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \sum_{t=1}^{M} \sum_{s=1}^{M} \mathrm{E}\left\{ c_t \cdot c_s \right\} - \mu_c^2 \right] \quad \text{(B.4.0.107)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(\sum_{t \neq s} \sum \mathrm{E}\left\{ c_t \cdot c_s \right\} + \sum_{t=1}^{M} \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.108)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(\sum_{t \neq s} \sum \mathrm{E}\left\{ c_t \right\} \cdot \mathrm{E}\left\{ c_s \right\} + \sum_{t=1}^{M} \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.109)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(\sum_{t \neq s} \sum (\mathrm{E}\left\{ c_t \right\})^2 + \sum_{t=1}^{M} \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.110)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(M \cdot (M-1) \cdot (\mathrm{E}\left\{ c_t \right\})^2 + M \cdot \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.111)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left((M^2 - M) \cdot (\mathrm{E}\left\{ c_t \right\})^2 + M \cdot \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.112)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(M^2 \cdot (\mathrm{E}\left\{ c_t \right\})^2 - M \cdot (\mathrm{E}\left\{ c_t \right\})^2 + M \cdot \mathrm{E}\left\{ c_t^2 \right\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.113)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(M^2 \cdot \mu_c^2 + M \cdot (\mathrm{E}\left\{ c_t^2 \right\} - (\mathrm{E}\left\{ c_t \right\})^2) \right) - \mu_c^2 \right] \quad \text{(B.4.0.114)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\frac{1}{M^2} \cdot \left(M^2 \cdot \mu_c^2 + M \cdot \sigma^2\{c_t\} \right) - \mu_c^2 \right] \quad \text{(B.4.0.115)}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\mu_c^2 + \frac{\sigma^2\{c_t\}}{M} - \mu_c^2\right] \tag{B.4.0.116}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[\left(\frac{\sigma\{c_t\}}{\sqrt{M}}\right)^2\right] \tag{B.4.0.117}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \left[(\sigma\{\bar{c}_t\})^2\right] \tag{B.4.0.118}$$

$$= \mathrm{E}\{y\} - \beta_1 \cdot (0) - \beta_2 \cdot \sigma^2\{\bar{c}_t\} \tag{B.4.0.119}$$

This shows the second order polynomial formulation of the standard CV, $(\bar{c} - \mu_c)$, is biased since $\mathrm{E}\{y\} \neq \mathrm{E}\{y_c\}$. The content that follows will make the contrast the formulation provided by [57] and Moy is not equivalent to equation (B.4.0.98) and contrary to the expected value of equation (B.4.0.98), their formulation is an unbiased estimate.

Moy [82, 83] formulates the quadratic polynomial CV as,

$$y_{cq} = y - g(c_1, c_2, \ldots, c_M) + \mathrm{E}\{g(c_1, c_2, \ldots, c_M)\}, \tag{B.4.0.120}$$

where

$$g(c_1, c_2, \ldots, c_M) = b_1 + b_2 \cdot \sum_{t=1}^{M} c_t + b_3 \cdot \left(\sum_{t=1}^{M} c_t\right)^2. \tag{B.4.0.121}$$

This formulation for y_{cq} is shown below to resolve to the original CV formulation y_c that is based on the evaluation of the average.

$$y_{cq} = y - g(c_1, c_2, \ldots, c_M) + \mathrm{E}\{g(c_1, c_2, \ldots, c_M)\} \tag{B.4.0.122}$$

$$= y - \left[b_1 + b_2 \cdot \sum_{t=1}^{M} c_t + b_3 \cdot \left(\sum_{t=1}^{M} c_t \right)^2 \right] + \mathrm{E} \left\{ \left[b_1 + b_2 \cdot \sum_{t=1}^{M} c_t + b_3 \cdot \left(\sum_{t=1}^{M} c_t \right)^2 \right] \right\} \tag{B.4.0.123}$$

$$= y - b_1 - b_2 \cdot \sum_{t=1}^{M} c_t - b_3 \cdot \left(\sum_{t=1}^{M} c_t \right)^2 + \mathrm{E}\{b_1\} + \mathrm{E} \left\{ b_2 \cdot \sum_{t=1}^{M} c_t \right\} + \mathrm{E} \left\{ b_3 \cdot \left(\sum_{t=1}^{M} c_t \right)^2 \right\} \tag{B.4.0.124}$$

$$= y - b_1 - b_2 \cdot \sum_{t=1}^{M} c_t - b_3 \cdot \left(\sum_{t=1}^{M} c_t \right)^2 + b_1 + b_2 \cdot \mathrm{E} \left\{ \sum_{t=1}^{M} c_t \right\} + b_3 \cdot \mathrm{E} \left\{ \left(\sum_{t=1}^{M} c_t \right)^2 \right\} \tag{B.4.0.125}$$

$$= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - \mathrm{E} \left\{ \sum_{t=1}^{M} c_t \right\} \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \mathrm{E} \left\{ \left(\sum_{t=1}^{M} c_t \right)^2 \right\} \right] \tag{B.4.0.126}$$

$$= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - \sum_{t=1}^{M} \mathrm{E} \left\{ c_t \right\} \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \mathrm{E} \left\{ \sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s \right\} \right] \tag{B.4.0.127}$$

$$= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - \sum_{t=1}^{M} \mu_c \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \sum_{t=1}^{M} \sum_{s=1}^{M} \mathrm{E} \left\{ c_t \cdot c_s \right\} \right] \tag{B.4.0.128}$$

$$= y - b_2 \cdot \left(\sum_{t=1}^{M} c_t - M \cdot \mu_c \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(\sum_{t \neq s} \sum \mathrm{E} \left\{ c_t \right\} \cdot \mathrm{E} \left\{ c_s \right\} + \sum_{t=1}^{M} \mathrm{E} \left\{ c_t^2 \right\} \right) \right] \tag{B.4.0.129}$$

$$= y - b_2 \cdot \left(M \cdot \left(\frac{1}{M} \cdot \sum_{t=1}^{M} c_t - \mu_c \right) \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(M \cdot (M-1) \cdot \left(\mathrm{E} \left\{ c_t \right\} \right)^2 + M \cdot \mathrm{E} \left\{ c_t^2 \right\} \right) \right] \tag{B.4.0.130}$$

$$= y - b_2 \cdot M \cdot \left(\sum_{t=1}^{M} \frac{c_t}{M} - \mu_c \right) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(M^2 \cdot \left(\mathrm{E} \left\{ c_t \right\} \right)^2 - M \cdot \left(\mathrm{E} \left\{ c_t \right\} \right)^2 + M \cdot \mathrm{E} \left\{ c_t^2 \right\} \right) \right] \tag{B.4.0.131}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(M^2 \cdot \mu_c^2 + M \cdot \left(\mathrm{E}\{c_t^2\} - (\mathrm{E}\{c_t\})^2 \right) \right) \right] \tag{B.4.0.132}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(M^2 \cdot \mu_c^2 + M \cdot \sigma^2\{c\} \right) \right] \tag{B.4.0.133}$$

$$\tag{B.4.0.134}$$

We extend the notation left off here by Kleijnen [57] .

$$y_{cq} = y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\left(\sum_{t=1}^{M} c_t \right)^2 - \left(M^2 \cdot \mu_c^2 + M \cdot \sigma^2\{c\} \right) \right] \tag{B.4.0.135}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s - M^2 \cdot \left(\mu_c^2 + \frac{\sigma^2\{c\}}{M} \right) \right] \tag{B.4.0.136}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s - M^2 \cdot \left(\mu_c^2 + \left(\frac{\sigma\{c\}}{\sqrt{M}} \right)^2 \right) \right] \tag{B.4.0.137}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot \left[\sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s - M^2 \cdot \left(\mu_c^2 + (\sigma\{\bar{c}\})^2 \right) \right] \tag{B.4.0.138}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot M^2 \cdot \left[\frac{1}{M^2} \sum_{t=1}^{M} \sum_{s=1}^{M} c_t \cdot c_s - \left(\mu_c^2 + \sigma^2\{\bar{c}\} \right) \right] \tag{B.4.0.139}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot M^2 \cdot \left[\sum_{t=1}^{M} \sum_{s=1}^{M} \frac{c_t \cdot c_s}{M^2} - \left(\mu_c^2 + \sigma^2\{\bar{c}\} \right) \right] \tag{B.4.0.140}$$

$$= y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot M^2 \cdot \left[(\bar{c})^2 - \left(\mu_c^2 + \sigma^2\{\bar{c}\} \right) \right] \tag{B.4.0.141}$$

Letting $\beta_1 = b_2 \cdot M$ and $\beta_2 = b_3 \cdot M^2$ gives

$$y_{cq} = y - b_2 \cdot M \cdot (\bar{c} - \mu_c) - b_3 \cdot M^2 \cdot \left[(\bar{c})^2 - \left(\mu_c^2 + \sigma^2\{\bar{c}\}\right)\right] \tag{B.4.0.142}$$

Now comparing equation (B.4.0.142) with equation (B.4.0.98), shows the far right term of both equations are the only elements in each equations that differ, $(\sigma^2\{\bar{c}\}) \neq (2 \cdot \bar{c} \cdot \mu_c)$. Finally, using the prior expected value results that showed the second order polynomial of the standard form of the CV, $(\bar{c} - \mu_c)$, was biased, and from the results that $\mathrm{E}\{(\bar{c})^2\} = (\mu_c^2 + \sigma^2\{\bar{c}\})$, then the expected value of equation (B.4.0.142) is unbiased since $\mathrm{E}\{y_{cq}\} = \mathrm{E}\{y\}$.

Bibliography

[1] Mohamed A. Ahmed, Donald Gross, and Douglas R. Miller. Control variate models for sensitivity estimates of repairable item systems. In *Proc. 19th Conf. Winter Simul. - WSC '87*, pages 324–333, New York, New York, USA, 1987. ACM Press.

[2] Fasihul M. Alam, Ken R. McNaught, and Trevor J. Ringrose. A comparison of experimental designs in the development of a neural network simulation metamodel. *Simul. Model. Pract. Theory*, 12(7-8):559–578, 2004.

[3] Abdulrahman Alenezi, Scott A. Moses, and Theodore B. Trafalis. Real-time prediction of order flowtimes using support vector regression. *Comput. Oper. Res.*, 35(11):3489–3503, 2008.

[4] Filippo Amato, José Luis González-Hernández, and Josef Havel. Artificial neural networks combined with experimental design: A soft approach for chemical kinetics. *Talanta*, 93:72–78, 2012.

[5] T.W. Anderson. *An Introduction to Multivariate Statistical Analysis.* Wiley-Interscience, 3rd edition, 2003.

[6] S. Andradottir. A review of simulation optimization techniques. In *1998 Winter Simul. Conf. Proc. (Cat. No.98CH36274)*, volume 1, pages 151–158. IEEE, 2013.

[7] Adedeji B. Badiru and David B. Sieger. Neural network as a simulation metamodel in economic analysis of risky projects. *Eur. J. Oper. Res.*, 105(1):130–142, 1998.

[8] Yukun Bao, Yansheng Lu, and Jinlong Zhang. Forecasting stock price by SVMs regression. pages 295–303. 2004.

[9] R.R. Barton. Simulation metamodels. In *1998 Winter Simul. Conf. Proc. (Cat. No.98CH36274)*, volume 1, pages 167–174. IEEE, 1998.

[10] Russell R. Barton. Metamodels for simulation input-output relations. *Proc. 24th Conf. Winter Simul. - WSC '92*, pages 289–299, 1992.

[11] Russell R. Barton and Martin Meckesheimer. Metamodel-based simulation optimization. In *Handbooks Oper. Res. Manag. Sci.*, volume 13, chapter 18, pages 535–574. 2006.

[12] Kenneth W. Bauer. *Control variate selection for multiresponse simulation.* PhD thesis, Purdue University, 1987.

[13] Kenneth W. Bauer, Stephen G. Alsing, and Kelly A. Greene. Feature screening using signal-to-noise ratios. *Neurocomputing*, 31(1-4):29–44, 2000.

[14] Kenneth W. Bauer, Sekhar Venkatraman, and James R. Wilson. Estimation procedures based on control variates with known covariance matrix. In *Proc. 19th Conf. Winter Simul. - WSC '87*, number 1985, pages 334–341, New York, New York, USA, 1987. ACM Press.

[15] Kenneth W. Bauer and James R. Wilson. Control-variate selection criteria. *Nav. Res. Logist.*, 39(3):307–321, 1992.

[16] Kenneth W. Bauer and James R. Wilson. Standardized routing variables: a new class of control variates. *J. Stat. Comput. Simul.*, 46(1-2):69–78, 1993.

[17] Mohktar S. Bazaraa, Hanif D. Sherali, and C.M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 3rd edition, 2006.

[18] J. Bellucci. *Non-Linear meta-modeling extensions to the robust parameter design of computer simulations*. PhD thesis, Air Force Institute of Technology, 2016.

[19] Sanjay K. Bose. *An Introduction to Queueing Systems*. Springer, 2002.

[20] G. E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. pages 270–310. 1992.

[21] Richard H. Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Math. Program.*, 89(1):149–185, 2000.

[22] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM J. Optim.*, 9(4):877–900, 1999.

[23] Byungwhan Kim and G.S. May. An optimal neural network process model for plasma etching. *IEEE Trans. Semicond. Manuf.*, 7(1):12–21, 1994.

[24] W. C. Carpenter and J. F. M. Barthelemy. A comparison of polynomial approximations and artificial neural nets as response surfaces. *Struct. Optim.*, 5(3):166–174, 1993.

[25] Russell C.H. Cheng. Analysis of simulation experiments under normality assumptions. *J. Oper. Res. Soc.*, 29(5):493–497, 1978.

[26] S.K Choudhury and G Bartarya. Role of temperature and surface finish in predicting tool wear using neural network and design of experiments. *Int. J. Mach. Tools Manuf.*, 43(7):747–753, 2003.

[27] RG Coyle. *System Dynamics Modelling: A Practical Approach*. Chapman and Hall/CRC, 1996.

[28] Kiran M. Desai, Shrikant A. Survase, Parag S. Saudagar, S.S. Lele, and Rekha S. Singhal. Comparison of artificial neural network (ANN) and response surface methodology (RSM) in fermentation media optimization: Case study of fermentative production of scleroglucan. *Biochem. Eng. J.*, 41(3):266–273, 2008.

[29] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2004.

[30] M. Farooq Anjum, Imran Tasadduq, and Khaled Al-Sultan. Response surface methodology: A neural network approach. *Eur. J. Oper. Res.*, 101(1):65–73, 1997.

[31] P. A. Fishwick. Neural network models in simulation: a comparison with traditional modeling approaches. In *Proc. 21st Conf. Winter Simul. - WSC '89*, pages 702–709, New York, New York, USA, 1989. ACM Press.

[32] Timothy D. Flietstra, Kenneth W. Bauer, and Jeffrey P. Kharoufeh. Integrated feature and architecture selection for radial basis neural networks. *Int. J. Smart Eng. Syst. Des.*, 5(4):507–516, 2003.

[33] Jay W Forrester. System dynamics, systems thinking, and soft OR. *Syst. Dyn. Rev.*, 10(2-3):245–256, 1994.

[34] Jay Wright Forrester. *Industrial Dynamics*. MIT Press, 1961.

[35] Linda W. Friedman. *The Simulation Metamodel*. Springer, 1st edition, 1996.

[36] Linda W. Friedman. *The Simulation Metamodel*. Springer Science & Business Media, 2nd edition, 2012.

[37] Peter W. Glynn and Ward Whitt. Indirect Estimation Via L = λ W. *Oper. Res.*, 37(1):82–103, 1989.

[38] Kelly A. Greene. *Feature saliency in artificial neural networks with application to modeling workload*. PhD thesis, Air Force Institute of Technology, 1998.

[39] Carl M. Gross, Donald; Shortle, John F.; Thompson, James M.; Harris. *Fundamentals of Queueing Theory*. Wiley-Interscience, 2008.

[40] Linda M Haines. Optimal design for neural networks. *New Dev. Appl. Exp. Des. - Sel. Proc. a 1997 Jt. AMS-ISM-SIAM Summer Conf.*, 34(January):152–162, 1998.

[41] J.M. Hammersley and D.C. Handscomb. *Monte Carlo Methods*, volume 13. Methuen & Co., London, 1964.

[42] S.S. Han, M. Ceiler, S.A. Bidstrup, P. Kohl, and G. May. Modeling the properties of PECVD silicon dioxide films using optimized back-propagation neural networks. *IEEE Trans. Components, Packag. Manuf. Technol. Part A*, 17(2):174–182, 1994.

[43] D.C. Handscomb. Variance reduction techniques. In *Des. Comput. Simul. Exp.* Duke University Press, Durham. NC, 1969.

[44] C.D. Himmel and G.S. May. Advantages of plasma etch modeling using neural networks over statistical techniques. *IEEE Trans. Semicond. Manuf.*, 6(2):103–111, 1993.

[45] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[46] R. D. Hurrion. Using a neural network to enhance the decision making quality of a visual interactive simulation model. *J. Oper. Res. Soc.*, 43(4):333–341, 1992.

[47] Ajaz S. Hussain, Xuanqiang Yu, and Robert D. Johnson. Application of neural computing in pharmaceutical product development. *Pharm. Res.*, 08(10):1248–1252, 1991.

[48] Donald L. Iglehart and Peter A.W. Lewis. Regenerative simulation with internal controls. *J. ACM*, 26(2):271–282, 1979.

[49] Sébastien Issanchou and Jean-Pierre Gauchi. Computer-aided optimal designs for improving neural network generalization. *Neural Networks*, 21(7):945–950, 2008.

[50] A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. Artificial neural networks: a tutorial. *Computer (Long. Beach. Calif).*, 29(3):31–44, 1996.

[51] J. Johnston and J. DiNardo. *Econometric Methods*. McGraw-Hill, 4th edition, 1996.

[52] Jae-Ryung Jung and Bong-Jin Yum. Artificial neural network based approach for dynamic parameter design. *Expert Syst. Appl.*, 38(1):504–510, 2011.

[53] Ronald C. Karnopp, Dean C.; Margolis, Donald L.; Rosenberg. *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. Wiley, 5th edition, 2012.

[54] Robert A. Kilmer. *Artificial neural network metamodels of stochastic computer simulations*. PhD thesis, University of Pittsburgh, 1994.

[55] Robert A. Kilmer, Alice E. Smith, and Larry J. Shuman. An emergency department simulation and a neural network metamodel. *J. Soc. Health Syst.*, 5(3):63–79, 1997.

[56] Sujin Kim and Shane G. Henderson. Adaptive control variates. In *Proc. 2004 Winter Simul. Conf. 2004.*, volume 1, pages 609–617. IEEE, 2004.

[57] Jack P. C. Kleijnen. Sensitivity analysis and optimization in simulation. In *Proc. 27th Conf. Winter Simul. - WSC '95*, pages 133–140, New York, New York, USA, 1995. ACM Press.

[58] Jack P.C. Kleijnen. *Statistical Techniques in Simulation Part I.* Dekker, 1st edition, 1974.

[59] Jack P.C. Kleijnen. Regression metamodels for generalizing simulation results. *IEEE Trans. Syst. Man. Cybern.*, 9(2):93–96, 1979.

[60] Jack P.C. Kleijnen. Regression metamodel summarization of model behaviour. Technical report, Department of Economics, Tilburg University, 1982.

[61] Jack P.C. Kleijnen. Experimental design for sensitivity analysis, optimization, and validation of simulation models. In Jerry Banks, editor, *Handb. Simul.*, pages 173–223. 1998.

[62] Jack P.C. Kleijnen. Kriging metamodeling in simulation: A review. *Eur. J. Oper. Res.*, 192(3):707–716, 2009.

[63] Jack P.C. Kleijnen and David Deflandre. Validation of regression metamodels in simulation: Bootstrap approach. *Eur. J. Oper. Res.*, 170(1):120–131, 2006.

[64] Jack P.C. Kleijnen, A.J. van den Burg, and R.Th. van der Ham. Generalization of simulation results practicality of statistical methods. *Eur. J. Oper. Res.*, 3(1):50–64, 1979.

[65] V.G. Kulkarni. *Introduction to Modeling and Analysis of Stochastic Systems.* Springer, 2011.

[66] Yiyo Kuo, Taho Yang, Brett A. Peters, and Ihui Chang. Simulation metamodel development using uniform design and neural networks for automated material handling systems in semiconductor wafer fabrication. *Simul. Model. Pract. Theory*, 15(8):1002–1015, 2007.

[67] Stephen S. Lavenberg, Thomas L. Moeller, and C. H. Sauer. Concomitant control variables applied to the regenerative simulation of queuing systems. *Oper. Res.*, 27(1):134–160, 1979.

[68] Stephen S. Lavenberg, Thomas L. Moeller, and Peter D. Welch. Statistical results on control variables with application to queueing network simulation. *Oper. Res.*, 30(1):182–202, 1982.

[69] Stephen S. Lavenberg and Peter D. Welch. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Manage. Sci.*, 27(3):322–335, 1981.

[70] A.M. Law. *Simulation Modeling and Analysis*. New York, New York, USA, 5th edition, 2014.

[71] M. Li, Y.F.; Ng S.; Xie. A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Appl. Soft Comput.*, 10(4):1257–1273, 2010.

[72] D. Y. Lin, B. M. Psaty, and R. A. Kronmal. Assessing the sensitivity of regression results to unmeasured confounders in observational studies. *Biometrics*, 54(3):948, 1998.

[73] P.L. Lunani, M.; Sudjianto, A.; Johnston. Generating efficient training samples for neural networks using Latin hypercube sampling. In *Intell. Syst. Through Artif. Neural Networks, Vol. 5*, pages 209–214. 1995.

[74] E Marengo, V Gianotti, S Angioi, and M.C Gennaro. Optimization by experimental design and artificial neural networks of the ion-interaction reversed-phase liquid chromatographic separation of twenty cosmetic preservatives. *J. Chromatogr. A*, 1029(1-2):57–65, 2004.

[75] Mathworks. Constrained Nonlinear Optimization Algorithms, 2016.

[76] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943.

[77] Joshua B. Meents. *Control variates and optimal designs in metamodeling*. PhD thesis, Air Force Institute of Technology, 2011.

[78] S. Minsky, M.; Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1988.

[79] Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley, New York, New York, USA, 8th edition, 2013.

[80] Douglas C. Montgomery and Gc Runger. *Applied statistics and probability for engineers*. John Wiley & Sons, 5th edition, 2010.

[81] Wiem Mouelhi-Chibani and Henri Pierreval. Training a neural network to select dispatching rules in real time. *Comput. Ind. Eng.*, 58(2):249–256, 2010.

[82] William A. Moy. *Sampling techniques for increasing the efficiency of simulations of queueing systems*. PhD thesis, Northwestern University, 1965.

[83] William A. Moy. Monte Carlo techniques: Practical. In *Des. Comput. Simul. Exp.*, pages 263–283. Duke University Press, 1969.

[84] Raymond H. Myers, Douglas C. Montgomery, and Christine M. Anderson-Cook. *Response Surface Methodology*. Wiley, 2009.

[85] Raymond H. Myers, Douglas C. Montgomery, G. Geoffrey Vining, and Timothy J. Robinson. *Generalized Linear Models with Applications in Engineering and the Sciences*. Wiley, 2nd edition, 2010.

[86] Barry L. Nelson. A perspective on variance reduction in dynamic simulation experiments. *Commun. Stat. - Simul. Comput.*, 16(2):385–426, 1987.

[87] Barry L. Nelson. On control variate estimators. *Comput. Oper. Res.*, 14(3):219–225, 1987.

[88] Barry L. Nelson. Control variate remedies. *Oper. Res.*, 38(6):974–992, 1990.

[89] Barry L. Nelson and Bruce W. Schmeiser. Decomposition of some well-known variance reduction techniques. Technical report, School of Industrial Engineering, Purdue University, 1984.

[90] Jose H. Noguera and Edward F. Watson. Response surface analysis of a multi-product batch processing facility using a simulation metamodel. *Int. J. Prod. Econ.*, 102(2):333–343, 2006.

[91] Klára Novotná, Jan Havliš, and Josef Havel. Optimisation of high performance liquid chromatography separation of neuroprotective peptides Fractional experimental designs combined with artificial neural networks. *J. Chromatogr. A*, 1096(1-2):50–57, 2005.

[92] Ardavan Nozari, Steven F. Arnold, and C. Dennis Pegden. Control variates for multipopulation simulation experiments. *IIE Trans.*, 16(2):159–169, 1984.

[93] Katsuhiko Ogata. *System Dynamics*. Pearson, 4th edition, 2004.

[94] M. Olazaran. A sociological study of the official history of the perceptrons controversy. *Soc. Stud. Sci.*, 26(3):611–659, 1996.

[95] Mark J.L. Orr. Introduction to radial basis function networks. Technical report, University of Edinburgh, Edinburgh, Scotland, 1996.

[96] M. L. Padgett and T. A. Roppel. Neural networks and simulation: Modeling for applications. *Simulation*, 58(5):295–304, 1992.

[97] William J. Palm. *System Dynamics*. McGraw-Hill, New York, NY, 2nd edition, 2005.

[98] H. Pierreval. Training a neural network by simulation for dispatching problems. In *Proc. Third Int. Conf. Comput. Integr. Manuf.*, pages 332–336. IEEE.

[99] Aćacio M. de O. Porta Nova and James R. Wilson. Using control variates to estimate multiresponse simulation metamodels. In *Proc. 18th Conf. Winter Simul. - WSC '86*, pages 326–334, New York, New York, USA, 1986. ACM Press.

[100] Aćacio M. de O. Porta Nova and James R. Wilson. Using control variates to estimate multiresponse simulation metamodels. In *Proc. 18th Conf. Winter Simul. - WSC '86*, pages 326–334, New York, New York, USA, 1986. ACM Press.

[101] Aćacio M. de O. Porta Nova and James R. Wilson. Estimation of multiresponse simulation metamodels using control variates. *Manage. Sci.*, 35(11):1316–1333, 1989.

[102] Aćacio M. de O. Porta Nova and James R. Wilson. Selecting control variates to estimate multiresponse simulation metamodels. *Eur. J. Oper. Res.*, 71(1):80–94, 1993.

[103] J. Prakash Maran, V. Sivakumar, K. Thirugnanasambandham, and R. Sridhar. Artificial neural network and response surface methodology modeling in mass transfer parameters predictions during osmotic dehydration of Carica papaya L. *Alexandria Eng. J.*, 52(3):507–516, 2013.

[104] A. Alan Pritsker. Introduction to simulation and SLAM II. 1986.

[105] F.W. Radema. *Variance reduction techniques in system simulation*. PhD thesis, Technische Hogeschool, Eindhoven, Netherlands, 1969.

[106] R. Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958.

[107] R. Frank Rosenblatt. Two theorems of statistical separability in the perceptron. Technical report, 1958.

[108] R. Frank Rosenblatt. Principles of neurodynamics. Perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc, Buffalo, NY, 1961.

[109] Reuven Y. Rubinstein and Ruth Marcus. Efficiency of multivariate control variates in monte carlo simulation. *Oper. Res.*, 33(3):661–677, 1985.

[110] R.J. Rumelhart, D.E.; Hinton, G.E.; Williams. Learning Internal Representations by Error Propagation. Technical report, UC San Diego, 1985.

[111] R.G. Sargent. Research issues in metamodeling. In *1991 Winter Simul. Conf. Proceedings.*, pages 888–893. IEEE, 1991.

[112] George A.F. Seber and Alan J. Lee. *Linear Regression Analysis.* Wiley, 2nd edition, 2003.

[113] Yichu Shan, Ruihuan Zhao, Yan Tian, Zhen Liang, and Yukui Zhang. Retention modeling and optimization of pH value and solvent composition in HPLC using back-propogation neural networks and uniform design. *J. Liq. Chromatogr. Relat. Technol.*, 25(7):1033–1047, 2002.

[114] Yness March Slokar, Jure Zupan, and Alenka Majcen Le Marechal. The use of artificial neural network (ANN) for modeling of the H2O2/UV decoloration process: part I. *Dye. Pigment.*, 42(2):123–135, 1999.

[115] Donald F. Specht. Probabilistic neural networks. *Neural Networks*, 3(1):109–118, 1990.

[116] J.M. Steppe, K.W. Bauer, and S.K. Rogers. Integrated feature architecture selection. *IEEE Trans. Neural Networks*, 7(4):1007–1014, 1996.

[117] James J. Swain. Control variates in nonlinear regression. In *Proc. - Winter Simul. Conf.*, pages 623–628, 1983.

[118] James J. Swain. Augmenting linear control variates using transformations. In *1989 Winter Simul. Conf. Proc.*, pages 455–458. IEEE, 1989.

[119] Roberto Szechtman. Control variates techniques for Monte Carlo simulation. In *Proc. 2003 Int. Conf. Mach. Learn. Cybern. (IEEE Cat. No.03EX693)*, volume 1, pages 144–149. IEEE, 2003.

[120] Jeffrey D. Tew. Using central composite designs in simulation experiments. In Intergovernmental Panel on Climate Change, editor, *Proc. 24th Conf. Winter Simul. - WSC '92*, volume 53, pages 529–538, New York, New York, USA, 1992. ACM Press.

[121] Jeffrey D. Tew and James R. Wilson. Estimating simulation metamodels using combined correlation-based variance reduction techniques. *IIE Trans.*, 26(3):2–16, 1994.

[122] Sekhar Venkatraman and James R. Wilson. The efficiency of control variates in multiresponse simulation. *Oper. Res. Lett.*, 5(1):37–42, 1986.

[123] R.A. Waltz, J.L. Morales, J. Nocedal, and D. Orban. An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Math. Program.*, 107(3):391–408, 2006.

[124] Philip D. Wasserman. *Advanced Methods in Neural Computing*. 1993.

[125] P.J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974.

[126] W. Whitt. The Queueing Network Analyzer. *Bell Syst. Tech. J.*, 62(9):2779–2815, 1983.

[127] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. *Proc. IEEE*, 78(9):1415–1442, 1990.

[128] James R. Wilson. Variance reduction: The current state. *Math. Comput. Simul.*, 25(1):55–59, 1983.

[129] James R. Wilson. Variance reduction In simulation. In *Proc. - Winter Simul. Conf.*, pages 123–128, 1984.

[130] James R. Wilson. Variance reduction techniques for digital simulation. *Am. J. Math. Manag. Sci.*, 4(3-4):277–312, 1984.

[131] James R. Wilson and Alan A.B. Pritsker. Experimental evaluation of variance reduction techniques for queueing simulation using generalized concomitant variables. *Manage. Sci.*, 30(12):1459–1472, 1984.

[132] James R. Wilson and Alan A.B. Pritsker. Variance reduction in queueing simulation using generalized concomitant variables. *J. Stat. Comput. Simul.*, 19(2):129–153, 1984.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
15–09–2016	PhD Dissertation	Oct 2013 – Sep 2016

4. TITLE AND SUBTITLE

ON IMPROVED LEAST SQUARES REGRESSION & ARTIFICIAL NEURAL NETWORK META-MODELS FOR SIMULATION VIA CONTROL VARIATES

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Michael P. Gibb, Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
Wright-Patterson AFB, OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT–ENS–DS–16–S–030

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Danny J. Huval
Air Force Nuclear Weapons Center
Bldg 20203
Kirtland AFB, NM 87117-5617
(505) 853-2024, danny.huval@kirtland.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFNWC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A. Approved for Public Release; Distribution Unlimited.

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States

14. ABSTRACT

The analysis of large-scale simulations can pose a large computational burden, often necessitating the use of high performance computers. Moreover, these models can be analytically intractable because of complex, internal logical relationships. To begin to overcome these types of obstacles, a method known as *meta-modeling* has been developed to construct mathematical functions that act as analytic surrogates to large scale simulations. This research examines the introduction of second-order interactions for two types of asymptotically-standardized linear control variates to least squares regression and radial basis neural network meta-models for a queuing simulation. Extensions are made to the statistical framework for variance reduction of direct estimation of single response, single population simulations and the framework for meta-models of single response, multiple population simulations. As a result, the new extensions are shown to significantly outperform existing frameworks and also provide the means to interpret and better understand the system dynamics that manifest when a system exhibits an exceptional amount of variance.

15. SUBJECT TERMS

Meta-model; Least Squares Regression; Artificial Neural Network; Radial Basis Function; Variance Reduction; System Dynamics; Control Variate; Design of Experiments; Jackson Open Network; Convex Non-linear Optimization; Simulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Kenneth W. Bauer, Jr., AFIT/ENS
U	U	U	UU	214	19b. TELEPHONE NUMBER *(include area code)* (937) 255–6565, x4328; kenneth.bauer@afit.edu